



Trading API Guide

Version 817

Contents

Copyright Notice	33
Release Notes	34
Welcome	35
Feedback and Support	35
Free Technical and Business Resources	35
Developer Technical Support	38
Other Contacts and Resources	39
About the Documentation	39
Intended Audience	40
How this Documentation is Organized	40
Related Documentation	41
Typographic Conventions	42
Diagram Conventions	43
Part I: Getting Started	44
Chapter 1: Product Overview	45
eBay Trading Web Services	45
eBay SDKs	46
Supported Use Cases	47
Types of Applications You Can Build	48
Listing Application	48
Seller Application	49
Gallery List Tool	49
API License Agreement and Usage	49
Chapter 2: System Requirements	50
Supported Client Environments	50
Where to Find the eBay Schema Files	51
URLs for the eBay Schema Files	51
Generating Interfaces and Proxies from a WSDL	52
URL for the API Gateway	52
Database Storage	53
Commonly Used Database Systems	53
Hardware Requirements	53
Next Steps	54

Chapter 3: Executing Your First API Call	55
Chapter 4: Developing an eBay-Enabled Application	56
Planning Ahead	56
Planning for the Compatible Application Check	57
Following Best Practices	58
Selecting Fields to Retrieve	59
Working with the Output Selector	60
Best Practices for the Output Selector	61
Testing Applications	61
About eBay Login Credentials	62
Using the Sandbox	62
Creating a Test User	63
Generating Auth Tokens for Test Users	65
Testing Your Application's API Access	66
Creating Stores in the Sandbox	67
Checklist for Going Live	67
Compatible Application Check	67
Supporting Compatibility Levels	71
Sandbox and Production Keys	71
Chapter 5: Invoking eBay Trading Web Services	74
Overview of the API Schema	75
Message Types	75
Core Components	78
Base Components	79
Release Versions	84
Interoperability Considerations	86
Routing the Request (Gateway URLs)	87
HTTP Headers (XML API Only)	87
SOAP URL Parameters (SOAP API and SDKs Only)	89
Standard Data for All Calls	92
Standard Input Data	92
Controlling the Amount of Data Returned	100
Security	101
Standard Output Data	104
Error Handling	107
Error Types	108
Customizing Your Application's Error Handling Abilities	113
Compatible Application Check Requirements for Error Handling	115
Getting Help When You Cannot Identify the Source of an Error	115
The Object Model of the eBay SDK for Java	115

eBay SDK for Java Libraries	115
Call Class Basics in the eBay SDK for Java	116
Chapter 6: Getting Tokens	117
Single-User Applications	117
Configuring the Consent Flow for Applications with Multiple Users	117
Setting Up an Application to Receive Tokens	118
Set Application-Level Settings	118
Generate an RuName for Your Application	119
Set RuName-Level Settings	120
Getting Tokens for Applications with Multiple Users	121
Option 1: Client/Desktop Applications	122
Option 2: Web/Server Applications	123
Preparing the URL for User Sign-in	125
The Form of the Accept Redirect URL	125
Getting a Token via FetchToken	125
Using Tokens	127
About Tokens	127
What is a Token?	127
Life Span and Uniqueness	128
Watching for Token Expiration	128
Watching for Token Expiration Causes	129
Token Expiration Causes	131
Part II: eBay Features Overview	132
Chapter 7: eBay Concepts	133
Basic Building Blocks	134
Summary of Building Blocks	134
Items and Listings	136
Categories (Introduction)	138
Choosing a Listing Option (Standard Way or with Product Details)	139
Users	145
Order Line Items	145
Feedback	145
eBay User Roles	147
Registered eBay Users	147
Application Developers	147
Sellers	148
Store Owners	148
Bidders and Buyers	148

eBay Sites and Environments	149
The Sandbox and Production Environments	149
The US and International eBay Sites	150
PayPal	153
eBay Motors	153
eBay Stores	154
Shopping.com	154
Half.com	155
Affiliate Tracking Concepts	155
Earn Commissions	155
Affiliate Tracking Enabled for Several Trading API Calls	156
Part III: Selling Items on eBay	157
Chapter 8: Categories	158
Retrieving the Category Hierarchy for a Site	158
Maintaining Category Data	161
Checking the Category Hierarchy Version	163
Mapping Old Category IDs to Current IDs	163
Determining Category Support for Certain Features	168
Determining Whether a Category Supports Reserve Prices	169
Determining Whether a Category Supports Item Specifics	170
Determining Whether a Category Supports Product Details	170
URLs for Category Lists for Global eBay Sites	170
Retrieving Suggested Categories Based on Keywords	171
Specifying a Query for Suggested Categories	172
Chapter 9: Listing Items	174
Listing Policies	174
Seller Limits	174
Listing an Item	176
Required Information for eBay Listings	177
Optional Information for eBay Listings	181
Considering Best Match When Listing Items	184
Considering Best Match When Revising and Relisting Items	184
Specifying a Payment Method	184
Determining the Payment Methods Allowed for a Category	185
Using Integrated Merchant Credit Card (IMCC)	186
Site Visibility	187
Requiring Immediate Payment	187
Requirements for Using Immediate Payment with AddItem	187
Testing the Immediate Payment Feature	188

Changing the Immediate Payment Requirement for an Item	189
Fees Resulting from Listing an Item	189
Fees Associated with Listing Features	189
Promotional Discounts	193
Chapter 10: Using PayPal	195
PayPal Overview	195
Offering PayPal Buyer Protection	196
Integrating PayPal With Your eBay Applications	197
Obtaining the PayPal ExternalTransactionID	197
The PayPal APIs	198
Using Instant Payment Notification	198
PayPal Transaction Reconciliation Report	199
Setting Up the PayPal Sandbox	199
Creating a PayPal Sandbox account	199
Logging in to the Sandbox	200
Using test accounts	200
Linking PayPal Test Account to eBay Sandbox	205
Chapter 11: Describing Items in a Listing	206
Introduction to Pictures in Item Listings	206
Working with Pictures in an Item Listing	208
Uploading Pictures to EPS	209
Associating Pictures with an Item	209
Using a Stock Photo in an Item Listing	210
Including Pictures in the Description and Using a Camera Icon	211
Retrieving Picture Data	212
Copy of a Self-Hosted Picture To eBay Picture Services	212
Photo Fees	213
Including Pictures in the Search Results Gallery	213
Gallery, Gallery Plus, and Gallery Featured	213
Using an eBay Stock Photo as a Gallery Picture	214
Using Gallery Features on Sites with Free Gallery	215
Specifying an Item's Condition	218
Using ConditionID to Specify an Item's Condition	218
Describing Details About the Condition of Items That Are Not New	
221	
Adding or Changing ConditionID When Revising or Relisting	222
Retrieving a Listing's Item Condition	223
Offering a Clear Return Policy	223
Listing an Item with a Return Policy	225
Retrieving an Item's Return Policy	225

Enabling Best Offer	225
Counteroffer Rules and Restrictions	226
Automatically Declining or Accepting an Offer	227
Anonymous Best Offer	228
Sellers' Use of Best Offer	230
Additional Help for Best Offer	231
Enabling Multi-jurisdiction Sales Tax	231
Overview	231
Using Tax Tables	232
How Tax Tables Are Applied	232
Enabling Get It Fast	233
Identifying Listings that Benefit Nonprofits	234
API Support for eBay Giving Works	234
Benefits of Retrieving Giving Works Information	236
Other Ways to List Charitable Items	236
Listing Items as Private	237
Adult Only Listings	238
Listing Adult Only Items	238
Retrieving Adult Items	238
Featuring Vehicles on the Mobile.de Site	239
Using a SKU Identifier with Items	239
Using Feature Packs to Save on Upgrade Fees	240
Making Listings Available by Default on Another Site	241
Prerequisites for the International Site Visibility Listing Upgrade	
242	
Chapter 12: DE/AT New eBay Payment Process	244
Setting Payout Preferences on Site	244
Changes to Listing Calls in Trading API	244
Specifying Payment Methods No Longer Required	244
Changes to Order Retrieval Calls in Trading API	245
Using GetOrders to Distinguish New Payment Process Orders	245
Retrieving eBay Seller Payout Information	245
Limitations of Order Update Calls	246
New Payment Process APIs	247
Payout Details API	247
Order Adjustment API	247
Unmatched EFT Payments	248
GetOrders Changes Related to Over/Under EFT Payments	248
Refunds	249
Refunds Through the Site	249
Refunds Through the Order Adjustment API	249

GetOrders Changes Related to Refunds	250
New Platform Notifications	250
ReadyToShip Notification	250
ReadyForPayout Notification	250
Unmatched EFT Payment Platform Notifications	251
Setting the API Compatibility Level	251
Categories Excluded from New Payment Process	251
Categories With Long Handling Time and/or Down Payments	252
Chapter 13: Using Description Templates	253
Layout Templates	254
Theme Templates	254
Theme Groups	255
Listing Designer Effects on the eBay API	256
Retrieving a List of Available Listing Designer Templates	257
Listing Items Using Listing Designer Templates	257
Retrieving Items that Have Assigned Listing Designer Templates	259
Displaying Item Descriptions with Layout Templates	259
Displaying Item Descriptions with Theme Templates	260
Revising and Relisting Items Using Listing Designer Templates	263
Chapter 14: Multi-Variation Listings	264
Overview of Multi-Variation Listings	265
Qualifications for Listing with Variations	267
Describing Variations in a Listing	268
Defining a Variation	269
Configuring Variation-Selection Widgets for Buyers (View Item)	
273	
Defining Pictures for Variations	275
Revising and Relisting with Variations	277
Changing the Category	277
Adding Variations	277
Deleting Individual Variations	277
Deleting All Variations	278
Modifying Existing Variations	278
Retrieving a Listing's Variations	280
Buying Items in Multi-Variation Listings	280
Retrieving Order Line Items with Variations	281
Monitoring Listings and Orders with Variations	281
Identifying Variations in Member Messages	281
Chapter 15: Listing Items with Parts Compatibility	282
Parts Compatibility Supported Categories	282

Checking Individual Leaf Categories	283
Identifying All Categories that Support Parts Compatibility	284
Parts Compatibility Listing Flow	285
Specifying Parts Compatibility by Product	286
Specifying Parts Compatibility Manually	289
Revising Parts Compatibilities for an Item	297
Retrieving Parts Compatibilities for Listed Items	299
Parts Compatibility Search Flow	302
Searching By Application	303
Searching By Specification	305
APIs for Parts Compatibility	306
Trading API	306
Product Services	307
Search API	308
Chapter 16: Want It Now	309
Retrieving Want It Now Posts	309
Searching for Posts with a Query	309
Searching for Posts by Category	310
Searching All eBay Sites for Posts	310
Paginating Search Results	310
Retrieving All Data for a Single Post	311
Responding to Want It Now Posts	311
Chapter 17: Shipping	312
Introduction to Shipping	312
Domestic and International Shipping	312
Shipping Carriers and Services	314
Shipping Types	315
Shipping Costs	316
Participation in the Global Shipping Program	317
Additional Resources	319
Working with Shipping in API Calls	319
Setting Versus Retrieving Shipping Details	319
Specifying Global Shipping in an Item Listing	320
Specifying Global Shipping-Related Item Attributes	320
Revising Shipping Details	321
Using Global Shipping Information After Checkout	322
Retrieving a Seller's Global Shipping Records	323
Specifying Locations to Where You Ship	324
Using Shipping Rate Tables for Finer Control of Shipping Location Charges	325

ShipToLocation(s)	325
Excluding Locations	326
US Listings and Package Size Limitations	326
Specifying Shipping Locations with the Global Shipping Program	
327	
Specifying Shipping Services	327
Required Domestic Shipping Service	327
Handling Time and Estimated Delivery Time	329
Enabling Get It Fast	331
Deprecated Shipping Services	334
Specifying Shipping Types and Costs	336
Mixing Shipping Types	336
Specifying Flat Rate Shipping	336
Specifying Calculated Shipping	341
Package Dimensions and Weight	342
Specifying Freight Shipping	346
UPS Rate Scales	349
Search Results and Shipping Costs	350
Specifying Other Costs	350
Packaging and Handling Cost	350
Shipping Surcharge	351
Taxes	351
Determining Shipping Costs for a Listing	352
Key Calls for Determining Shipping Costs	352
Determining Flat Rate Shipping Costs	353
Determining Calculated Shipping Costs	354
Determining Freight Shipping Cost	355
Other Shipping Features	355
Ensuring PayPal Seller Protection	355
Shipping Cost Discount Profiles	356
Specifying the Cash on Delivery Option in Shipping Details	361
Shipping Insurance	362
Chapter 18: Specialty Categories and Listings	366
Classified Ad Listings	366
eBay Real Estate Listings	367
Digital (Downloadable File) Listings	368
Ticket Listings	370
Basic Ticket Listing Requirements	371
Listing Tickets with Product Details	371
Filling in All Ticket Item Specifics Manually	373

Part IV: Managing Listed Items	374
Chapter 19: Managing Item Listings	375
Relisting Items	376
General Steps	376
Indicating Changed Fields When Relisting an Item	377
Removing a Field When Relisting an Item	377
Changing Payment Methods in a Relisted Item	378
Relisting Items with Product Details (Catalogs)	379
Removing a Listing Subtitle When Relisting an Item	380
Relisting and Versions	380
Relisting Eligible Administratively Cancelled Items	380
Using VerifyRelistItem to test RelistItem	381
Revising Items	381
Qualifications to Revise an Item	382
Using the VerifyOnly Field to Test Your Changes	382
Item Changes Allowed During Revisions	382
Revising eBay Giving Works (Charity) Listings	383
Revising eBay Stores Listings	383
Revising Items After Purchases are Made	383
Removing a Field When Revising an Item	383
Changing Payment Methods When Revising an Item	384
Revising Multiple-Choice (Unbounded) Fields	384
Revising US eBay Motors Listings	385
Revising Real Estate Listings	386
Revising Items with Product Details (Catalogs)	386
Changing a Listing Start Price	388
Reducing Reserve Price	388
Changing Item Descriptions	388
Ending Items Early	389
Working with Seller Events	391
Learning About Bidders	393
Putting Store Items on Sale	394
Promotional Sale API	394
Defining a Promotional Sale	394
Managing a Promotional Sale	396
Promotional Sale Rules	396
Markdown Manager	397
Using Selling Manager Calls in the Trading API	397
Subscribing a User to Selling Manager	397
Concepts for Selling Manager Calls	398

Retrieving Selling Manager Inventory and Templates	398
Adding, Revising, and Deleting Products and Templates	399
Using Selling Manager Automation Rules	399
Retrieving Selling Manager Alerts and Email Logs	399
Retrieving and Modifying Data About Sold Items	400
Selling Manager Documentation	400
Chapter 20: Communication Between Members	402
Summary of Messaging Calls	402
Static Email Addresses in Trading API Calls	402
Sending Messages	403
Messages Sent Regardless of Relationship	403
Messages Sent When an Order Relationship Exists	404
If the Users Have a Bidding Relationship	404
Retrieving Messages	404
Example: Responding to a Question	404
Enabling Communication Through Skype	405
Making Skype Available In Listings	405
Changes in Skype Functionality	406
Chapter 21: Checkout	407
Creation of an Order Line Item	407
Listing Fields and Features Affecting an Order Line Item	408
Reviewing and Modifying an Order Line Item	411
Review Your Purchase Page	411
Buyer and Seller Communication Prior to Payment	411
Revising Checkout Details	411
Creating a Combined Payment Order	412
Buyer Payment	414
Chapter 22: Completing the Sale	415
Retrieving Order Line Item Data and Managing Orders	416
Retrieving Order Line Item Data Directly	416
Best Practice for GetOrders	417
Subscribing to Platform Notifications	417
Retrieving the Order Line Items for a Single Item	418
Retrieving the Order Line Items for a Specific Seller	419
Understanding Buyer Payment	420
Holds on PayPal Payments	421
Determining the Hold Status	421
Payment Review Hold	422
Merchant Hold	423
New Seller Hold	423

Handling Non-Paying Bidders	424
CompleteSale API Call	424
Shipping the Item	424
Final Value Fees and Credits	425
Leaving and Getting Feedback	426
Making Second Chance Offers for Items	426
Supporting API Calls	427
Determining Item Eligibility for a Second Chance Offer	427
Extending a Second Chance Offer	429
Retrieving Item Data for Second Chance Offer Listings	430
Terminating a Second Chance Offer	431
Chapter 23: Leaving and Retrieving Feedback	432
Introduction to Feedback	432
Detailed Seller Ratings	433
Using the API for Detailed Seller Ratings	434
Leaving Feedback for Another User	435
Getting Feedback Left By Another User	436
Making the Call	437
Working with the GetFeedback Result Set	437
Responding to Feedback	440
Chapter 24: Disputes Overview	441
Resolving Selling Issues	441
Resolving Buying Issues	442
eBay Buyer Protection Program	443
Legacy eBay Dispute System	444
PayPal Buyer Protection	444
Overview of Dispute Actions	444
Disputes and the Resolution Case Management API	446
Overview	446
Using Case Retrieval Calls	448
Using Filters With the getUserCases Call	449
Retrieving a List of Next Possible Actions on a Case	450
Requesting the Return of a SNAD Item	451
Issuing a Full Refund to the Buyer	451
Offering a Partial Refund to the Buyer	451
Issuing a Partial Refund to the Buyer	451
Providing Shipping Information to the Buyer	452
Providing Tracking Information to the Buyer	452
Providing Return Information to the Buyer	452
Providing Refund Information to the Buyer	452

Uploading Proof Documents	452
Offering Other Solution	453
Escalating a Case	453
Appealing a Case	453
eBay Buyer Protection Platform Notifications	454
Chapter 25: Unpaid Item Disputes	455
Creating Unpaid Item Disputes in the Resolution Center	456
Canceling a Transaction in the Resolution Center	456
Creating and Managing Disputes With Trading API	457
Using AddDispute	457
Notifications	458
Unpaid Item Disputes Process	458
Unpaid Item Dispute Diagram for Version 637 or Later	460
Unpaid Item Dispute Diagram for Versions Earlier Than 637	461
Unpaid Item Dispute States and Consequences	463
Unpaid Item Assistant	463
Introduction	463
Unpaid Item Assistant Preferences	464
API Support for Unpaid Item Assistant	464
Communicating with the Buyer	466
Version 637 or Greater	466
Prior to Version 637	466
Closing an Unpaid Item Dispute	467
If the Buyer and Seller Complete the Transaction	468
If the Seller Chooses to Not Complete the Transaction	468
If the Seller Decides to End Communication with the Buyer	468
Notifications	469
Getting Details About a Dispute	469
Overview	469
GetDispute	469
GetUserDisputes	470
Reversing a Closed Unpaid Item Dispute	470
Overview	470
SellerReverseDispute	470
Testing Unpaid Item Disputes in the Sandbox	471
Notifications Related to the Unpaid Item Process	473
DisputeState and DisputeActivity for Unpaid Items	473
Implications of Various Values of DisputeActivity	474
Chapter 26: Buyer Disputes	476
Buyer Disputes on US, UK and DE Sites	476

Buyer Disputes on Other International Sites	477
Filing Buyer Disputes on PayPal	477
Responding to and Tracking Buyer Disputes Through Trading API	477
General Flow for Buyer Disputes	478
Part V: Item Specifics and Catalog Products	480
Chapter 27: Working with Custom Item Specifics	481
About Item Specifics	481
Retrieving a Listing's Item Specifics	483
Searching Against Custom Item Specifics	484
Pre-filling Item Specifics with Product Details	484
ProductID Error Handling	486
Checking Category Support for Custom Item Specifics	487
Listing with Custom Item Specifics	488
Revising and Relisting with Custom Item Specifics	489
Using Advanced Features of Custom Item Specifics	490
Listing in Two Categories with Custom Item Specifics	490
Listing with Recommended Item Specific Names and Values	490
Listing with Required Item Specifics	495
Checking Valid String Lengths for Custom Item Specifics	495
Using Standard Units of Measure for Custom Item Specifics	496
Retrieving Recommendations for All Categories on a Site	496
Part VI: Searching for and Retrieving Items	497
Chapter 28: Searching for Items	498
Searching for Items by Using a Query	498
Searching for Reviews & Guides	499
Searching for Products, Reviews, and Guides	499
Presenting Links to Buying Guides	500
Finding All Item Listings in a Specified Category	501
Chapter 29: Retrieving Items	502
Browsing a Seller's Items	502
Retrieving the Data for One Item	504
Retrieving an Item with Item Specifics and/or Product Data	504
Displaying Discount Pricing Information to Buyers	505
Display Guidelines for Strikethrough Pricing	505
Display Rules for Minimum Advertised Price	510
Additional Discount Pricing Information	511

Part VII:Retrieving User Data	512
Chapter 30: Managing User Information	513
Retrieving User Information	513
Uniquely Identifying a User	514
Working with Anonymous User Information	515
Managing User Preferences	517
Getting User Preferences	517
Setting User Preferences	517
Using the Seller Dashboard to Monitor Your Status	520
Getting Account Information	521
Generating an Invoice Report	522
Generating a Period Report	522
Working with the GetAccount Result Set	523
Getting Information about a User's Store	524
Email and Address Privacy Policy	524
Email Privacy Policy	524
RegistrationAddress Privacy Policy	524
Chapter 31: My eBay	525
My eBay Overview	525
Replicating My eBay in an Application	526
Replicating My eBay Views with the API	526
Replicating Specific My eBay Views with the API	527
Retrieving Items with GetMyeBayBuying	529
Retrieving Items with GetMyeBaySelling	531
Part VIII:eBay Stores	533
Chapter 32: Managing eBay Stores	534
Creating a Store	535
Editing the Store Settings	536
Managing Store Categories	539
Managing Custom Pages	540
Managing a User's Store Preferences	542
Setting Store Preferences	542
Getting Store Preferences	542
Special Listing Fields for eBay Stores Sellers	543
Listing US eBay Motors Items in eBay Stores	543
Chapter 33: Cross-Promotions	544
Overview	544
Getting Set Up to Use Merchandising Manager	545
Cross-Promotion Methods	545

Cross-Promotion Rule Schemes	546
Query Filters	547
Types of Rules	548
Cross-Promotion Placements	549
Cross-Promotion Presentation Styles	551
Defining Cross-Promotion Rules	553
Retrieving Cross-Promotion Rules	554
Retrieving Cross-Promoted Items for Display with a Referring Item	
554	
Related Topics	555
Part IX: Specialty and International Sites	556
Chapter 34: Working with Special eBay Sites	557
Working with Features for eBay Stores Sellers	557
Working with eBay Real Estate Listings	557
Working with US and CA eBay Motors Listings	558
Working with Half.com Listings	558
Chapter 35: eBay Motors	559
Listing US and CA eBay Motors Items	559
Specifying Categories for eBay Motors Listings	560
Specifying a Listing Format for eBay Motors Listings	561
eBay Motors and eBay Stores	561
Shipping Details for eBay Motors	562
Local Market Listings	562
Specifying the Vehicle Identification Number (VIN)	564
Using Item Specifics in eBay Motors US Listings	565
Specifying the Vehicle Manufacturer and Model	567
Specifying a Subtitle for a Motors Vehicle Listing	568
Payment Methods for Motors	568
Specifying a Vehicle Deposit and Deposit Amount	568
Requiring Immediate Payment of a Vehicle Deposit	569
Vehicle Purchase Protection	570
Listing eBay Motors US Items in eBay Stores	570
Revising US Motors Listings	570
Short-Term Service Agreement (Limited Warranty)	570
Searching for eBay Motors US Listings	571
Working with eBay Motors Germany Listings	572
Cross-Promoting Vehicles on the Mobile.de Site	572
Working with eBay Motors Australia Listings	573
Specifying the VIN and Registration Date for eBay Australia	573

Chapter 36: Half.com	574
The Half.com Listing Workflow	574
How to Become a Half.com Seller	575
Half.com Payments and Commissions	575
Half.com Pay Periods and Payments	575
Applicable Fields for Half.com Listings	577
Applicable Calls for Half.com	579
Chapter 37: International Differences Overview	582
Chapter 38: Working with Internationalization Issues	584
Internationalization of the API	584
Effects of Site IDs	585
Character Encoding	585
Chapter 39: Working with Business Features and VAT	586
Value-Added Tax (VAT)	586
VAT-exempt Sellers	586
Advantages of Being VAT-exempt	587
Business Features on German Sites	587
Advantages of Being a B2B Seller	587
Adding a Business Item	588
Modifying Business Items	588
Retrieving Information about Business Users and Items	589
Chapter 40: Field Differences for eBay Sites	590
Business Feature Field Differences	591
Buyer Requirements, Verified User (VerifiedUser)	591
Currency Codes	593
Fees per Site	593
Hit Counter Differences (HitCounter)	595
Listing Duration Logic Differences	596
Minimum Feedback Requirements for Various Features	597
Miscellaneous Item Field Differences	600
Payment Method Differences (PaymentMethod)	601
Postal and Zip Codes (PostalCode)	603
Real Estate Ad Format Differences	604
Search Type Differences	605
Seller Level Differences (SellerLevel)	607
Seller Payment Methods (SellerPaymentMethod)	608
Shipping Region and Shipping Option Differences	609
Promotional Sale Differences	610
Site IDs	611
Site ID Filters for Search Calls (SiteIdFilter)	613

Tax Field Differences (SalesTax, UseTaxTable)	614
Tax Jurisdictions	615
Sites for which There are Currently no Tax Jurisdictions	615
CA (2)	616
IN (2)	616
US (0)	618
Part X: Platform Notifications Guide	620
Chapter 41: Working with Platform Notifications	623
Notifications for Buyers and Sellers	624
Subscribing to Platform Notifications	624
Basic Requirements	624
Notifications for Fixed-Price Listings with Variations or Multiple Quantities	625
Receiving Platform Notifications	625
Responding to Notifications	626
SOAP Message Fields	626
Notification Payloads	628
Receiving Notifications by Email	630
Informational Alerts	631
Account Activity Summary	631
Chapter 42: Wireless Device Applications	634
Support for Wireless Applications	634
Notification Events for Wireless	634
Chapter 43: AskSellerQuestion	636
Platform Choice	636
Data Fields Returned	636
Example	637
Chapter 44: AuctionCheckoutComplete	638
Data Fields Returned	638
Example	639
Chapter 45: BestOffer	642
Data Fields Returned	642
Example	643
Chapter 46: BestOfferDeclined	644
Data Fields Returned	644
Example	645
Chapter 47: BestOfferPlaced	646
Data Fields Returned	646

Example	647
Chapter 48: BidPlaced	648
Data Fields Returned	648
Example	648
Chapter 49: BidReceived	651
Data Fields Returned	651
Example	651
Chapter 50: BulkDataExchangeJobCompleted	656
Platform Choice	656
Data Fields Returned	656
Example	657
Chapter 51: Checkout	658
Example	660
Chapter 52: CounterOfferReceived	663
Data Fields Returned	663
Example	664
Chapter 53: CheckoutBuyerRequestsTotal	665
Data Fields Returned	665
Chapter 54: eBay Buyer Protection Notifications	666
eBP Notification Structure	667
EBPMyResponseDue	667
EBPOtherPartyResponseDue	668
EBPEscalatedCase	668
EBPAppealedCase	668
EBPMyPaymentDue	668
EBPMyPaymentDone	668
EBPClosedAppeal	669
EBPClosedCase	669
EBPOnHoldCase	669
Chapter 55: EndOfAuction	670
Platform Choice	670
Data Fields Returned	670
Example	671
Chapter 56: Feedback Notifications	675
Feedback	675
Data Fields Returned	675
Example	676
Chapter 57: FeedbackLeft	677
Data Fields Returned	677

Example	678
Chapter 58: FeedbackReceived	679
Data Fields Returned	679
Example	680
Chapter 59: FeedbackStarChanged	681
Data Fields Returned	681
Example	682
Chapter 60: FixedPriceTransaction	684
Data Fields Returned	684
Chapter 61: INR (ItemNotReceived) Notifications	685
INRBuyerOpenedDispute	685
Data Fields Returned	685
INRBuyerRespondedToDispute	685
Data Fields Returned	685
INRBViewerClosedDispute	686
Data Fields Returned	686
INRSellerRespondedToDispute	686
Data Fields Returned	686
Example	686
Chapter 62: ItemAddedToBidGroup	688
Data Fields Returned	688
Example	689
Chapter 63: ItemAddedToWatchList	691
Data Fields Returned	691
Example	691
Chapter 64: ItemClosed	694
Data Fields Returned	694
Example	695
Chapter 65: ItemListed	699
Data Fields Returned	699
Example	700
Chapter 66: ItemLost	703
Data Fields Returned	703
Example	704
Chapter 67: ItemMarkedPaid	707
Data Fields Returned	707
Example	708

Chapter 68: ItemMarkedShipped	711
Data Fields Returned	711
Example	712
Chapter 69: ItemRemovedFromBidGroup	715
Platform Choice	715
Data Fields Returned	715
Example	716
Chapter 70: ItemRemovedFromWatchList	718
Data Fields Returned	718
Example	719
Chapter 71: ItemRevised	722
Data Fields Returned	722
Example	723
Chapter 72: ItemSold	725
Data Fields Returned	725
Example	726
Chapter 73: ItemUnsold	729
Data Fields Returned	729
Example	730
Chapter 74: ItemWon	733
Data Fields Returned	733
Example	734
Chapter 75: MyMessages Notification	737
Setting MyMessages Notification Preferences	737
Subscribing to a MyMessages Notification	738
Subscribing to a Different Amount of Detail	738
MyMessagesAlert	738
MyMessagesAlertHeader	738
Example	738
MyMessageseBayMessage	739
MyMessageseBayMessageHeader	739
Example	740
MyMessagesM2MMMessage	740
MyMessagesM2MMMessageHeader	740
Chapter 76: OutBid	742
Platform Choice	742
Data Fields Returned	742
Example	743

Chapter 77: eBay Return Notifications	746
Return Notification Structure	747
ReturnCreated	747
ReturnWaitingForSellerInfo	748
ReturnSellerInfoOverdue	748
ReturnShipped	748
ReturnDelivered	748
ReturnRefundOverdue	748
ReturnClosed	748
ReturnEscalated	749
Chapter 78: SecondChanceOffer	750
Platform Choice	750
Data Fields Returned	750
Chapter 79: BuyerResponseDispute	751
Data Fields Returned	751
Example	752
Chapter 80: SellerClosedDispute	754
Data Fields Returned	754
Example	754
Chapter 81: SellerOpenedDispute	756
Data Fields Returned	756
Example	756
Chapter 82: Seller Payout Notifications	758
ReadyToShip	758
ReadyForPayout	759
UnmatchedPaymentReceived	759
RefundSuccess	759
RefundFailure	759
Chapter 83: SellerRespondedToDispute	760
Data Fields Returned	760
Example	760
Chapter 84: TokenRevocation	762
Data Fields Returned	762
Example	762
Chapter 85: WatchedItemEndingSoon	763
Platform Choice	763
Data Fields Returned	763
Example	764

Part XI: API Call Reference Guide	765
Chapter 86: Overview of Calls	770
Chapter 87: AddDispute	771
AddDispute Input Fields	771
AddDispute Output Fields	771
AddDispute Details	772
Working with the Response	772
AddDispute Best Practices	772
Managing the Volume of the Response	772
Testing AddDispute	772
AddDispute Sample	773
Related Calls	773
Change History	774
Chapter 88: AddDisputeResponse	775
Overview	775
Unpaid Item Disputes	775
Item Not Received Disputes	775
AddDisputeResponse Input Fields	776
AddDisputeResponse Output Fields	777
AddDisputeResponse Details	777
Prerequisites	777
Working with the Response	777
AddDisputeResponse Best Practices	777
Managing the Volume of the Response	777
Reducing the Number of Calls You Make	778
Testing AddDisputeResponse	778
AddDisputeResponse Samples	778
Unpaid Item Dispute	779
Item Not Received Dispute	779
Related Calls	780
Change History	780

Chapter 89: AddFixedPriceItem	781
Chapter 90: AddItem	782
Chapter 91: AddItems	783
Chapter 92: AddMemberMessageAAQToPartner	784
Chapter 93: AddMemberMessageRTQ	785
Chapter 94: AddMemberMessagesAAQToBidder	786
Chapter 95: AddOrder	787
Chapter 96: AddSecondChanceItem	788
Chapter 97: AddToItemDescription	789
Chapter 98: AddToWatchList	790
Chapter 99: AddTransactionConfirmationItem	791
Chapter 100: CompleteSale	792
Chapter 101: ConfirmIdentity	793
Chapter 102: DeleteMyMessages	794
Chapter 103: EndFixedPriceItem	795
Chapter 104: EndItem	796
Chapter 105: EndItems	797
Chapter 106: FetchToken	798
Chapter 107: GetAccount	799
Chapter 108: GetAdFormatLeads	800
Chapter 109: GetAllBidders	801
Chapter 110: GetApiAccessRules	802
Chapter 111: GetBestOffers	803
Chapter 112: GetBidderList	804
Chapter 113: GetCategories	805
Chapter 114: GetCategory2CS	806
Chapter 115: GetCategory2FinanceOffer	807
Chapter 116: GetCategoryFeatures	808
Chapter 117: GetCategoryListings	809
Chapter 118: GetCategoryMappings	810
Chapter 119: GetCategorySpecifics	811
Chapter 120: GetChallengeToken	812
Chapter 121: GetCharities	813
GetCharities Input Fields	813

Valid CharityDomain Values	813
Valid CharityRegion Values	815
GetCharities Output Fields	820
GetCharities Best Practices	820
Managing the Volume of the Response	820
Testing GetCharities	821
GetCharities Sample	821
Related Calls	822
Change History	822
Chapter 122: GetClientAlertsAuthToken	824
Chapter 123: GetContextualKeywords	825
Chapter 124: GetCrossPromotions	826
Chapter 125: GetDescriptionTemplates	827
Chapter 126: GetDispute	828
Chapter 127: GeteBayDetails	829
Chapter 128: GeteBayOfficialTime	830
Chapter 129: GetFeedback	831
Chapter 130: GetFinanceOffers	832
Chapter 131: GetItem	833
Chapter 132: GetItemsAwaitingFeedback	834
Chapter 133: GetItemShipping	835
Chapter 134: GetItemTransactions	836
Chapter 135: GetMemberMessages	837
Chapter 136: GetMessagePreferences	838
Chapter 137: GetMyeBayBuying	839
Chapter 138: GetMyeBayReminders	840
Chapter 139: GetMyeBaySelling	841
Chapter 140: GetMyMessages	842
Chapter 141: GetNotificationPreferences	843
GetNotificationPreferences Input Fields	843
GetNotificationPreferences Output Fields	844
GetNotificationPreferences Details	844
Prerequisites	844
Working with the Response	844
GetNotificationPreferences Best Practices	844
Managing the Volume of the Response	844
Reducing the Number of Calls You Make	845

Testing GetNotificationPreferences	845
GetNotificationPreferences Samples	845
Related Calls	848
Change History	848
Chapter 142: GetNotificationsUsage	849
GetNotificationsUsage Samples	849
Related Calls	851
Change History	851
Chapter 143: GetOrders	852
Chapter 144: GetOrderTransactions	853
Chapter 145: GetPromotionalSaleDetails	854
Chapter 146: GetPromotionRules	855
Chapter 147: GetSearchResults	856
Chapter 148: GetSellerDashboard	857
Chapter 149: GetSellerEvents	858
Chapter 150: GetSellerList	859
Chapter 151: GetSellerPayments	860
GetSellerPayments Input Fields	860
GetSellerPayments Output Fields	860
GetSellerPayments Details	861
Prerequisites	861
Working with the Response	861
GetSellerPayments Best Practices	861
Managing the Volume of the Response	861
Reducing the Number of Calls You Make	861
GetSellerPayments Samples	862
Related Calls	863
Change History	863

Chapter 152: GetSellerTransactions	865
Chapter 153: GetSessionID	866
Chapter 154: GetShippingDiscountProfiles	867
Chapter 155: GetStore	868
Chapter 156: GetStoreCategoryUpdateStatus	869
Chapter 157: GetStoreCustomPage	870
Chapter 158: GetStoreOptions	871
Chapter 159: GetStorePreferences	872
Chapter 160: GetSuggestedCategories	873
Chapter 161: GetTaxTable	874
Chapter 162: GetTokenStatus	875
Chapter 163: GetUser	876
Chapter 164: GetUserContactDetails	877
Chapter 165: GetUserDisputes	878
Chapter 166: GetUserPreferences	879
Chapter 167: GetVeROReasonCodeDetails	880
GetVeROReasonCodeDetails Input Fields	880
GetVeROReasonCodeDetails Output Fields	880
GetVeROReasonCodeDetails Sample	880
GetVeROReasonCodeDetails Details	882
Prerequisites	882
GetVeROReasonCodeDetails Best Practices	882
Managing the Volume of the Response	882
Testing GetVeROReasonCodeDetails	883
Related Calls	883
Change History	883
Chapter 168: GetVeROReportStatus	884
GetVeROReportStatus Input Fields	884
GetVeROReportStatus Output Fields	884
GetVeROReportStatus Samples	885
GetVeROReportStatus Details	887
Prerequisites	887
GetVeROReportStatus Best Practices	887
Managing the Volume of the Response	887
Testing GetVeROReportStatus	887
Related Calls	888
Change History	888

Chapter 169: GetWantItNowPost	889
GetWantItNowPost Input Fields	889
GetWantItNowPost Output Fields	889
GetWantItNowPost Details	890
Prerequisites	890
Working with the Response	890
GetWantItNowPost Best Practices	890
Managing the Volume of the Response	890
Reducing the Number of Calls You Make	890
Testing GetWantItNowPost	890
GetWantItNowPost Sample	891
Related Calls	892
Change History	892
Chapter 170: GetWantItNowSearchResults	893
GetWantItNowSearchResults Input Fields	893
GetWantItNowSearchResults Output Fields	894
GetWantItNowSearchResults Details	894
Prerequisites	894
Working with the Response	894
GetWantItNowSearchResults Best Practices	895
Managing the Volume of the Response	895
Testing GetWantItNowSearchResults	895
GetWantItNowSearchResults Samples	896
Search by Category	896
Keyword Query	897
Related Calls	898
Change History	899
Chapter 171: IssueRefund	900
IssueRefund Input Fields	900
IssueRefund Output Fields	900
IssueRefund Details	901
Prerequisites	901
Working with the Response	901
IssueRefund Best Practices	901
Managing the Volume of the Response	901
Reducing the Number of Calls You Make	901
IssueRefund Samples	902
Related Calls	902
Change History	902

Chapter 172: LeaveFeedback	903
Chapter 173: PlaceOffer	904
Chapter 174: RelistFixedPriceItem	905
Chapter 175: RelistItem	906
Chapter 176: RemoveFromWatchList	907
Chapter 177: RespondToBestOffer	908
RespondToBestOffer Input Fields	908
RespondToBestOffer Output Fields	908
RespondToBestOffer Samples	908
RespondToBestOffer Details	910
RespondToBestOffer Best Practices	910
Managing the Volume of the Response	910
Reducing the Number of Calls You Make	911
Testing RespondToBestOffer	911
Related Calls	911
Change History	911
Chapter 178: RespondToFeedback	912
Chapter 179: RespondToWantItNowPost	913
RespondToWantItNowPost Input Fields	913
RespondToWantItNowPost Output Fields	913
RespondToWantItNowPost Details	914
Prerequisites	914
Working with the Response	914
RespondToWantItNowPost Best Practices	914
Managing the Volume of the Response	914
Reducing the Number of Calls You Make	914
Testing RespondToWantItNowPost	914
RespondToWantItNowPost Sample	914
Related Calls	915
Change History	915
Chapter 180: ReviseCheckoutStatus	916
Chapter 181: ReviseFixedPriceItem	917
Chapter 182: ReviseItem	918
Chapter 183: ReviseMyMessages	919
Chapter 184: ReviseMyMessagesFolders	920
Chapter 185: RevokeToken	921
Chapter 186: SellerReverseDispute	922
Overview	922

SellerReverseDispute Input Fields	922
SellerReverseDisputeCall Output Fields	922
SellerReverseDispute Best Practices	923
Managing the Volume of the Response	923
Testing SellerReverseDispute	923
SellerReverseDispute Samples	923
Related Calls	925
Change History	925
Chapter 187: SendInvoice	927
Chapter 188: SetMessagePreferences	928
Chapter 189: SetNotificationPreferences	929
Chapter 190: SetPromotionalSale	930
Chapter 191: SetPromotionalSaleListings	931
Chapter 192: SetPromotionRules	932
Chapter 193: SetShippingDiscountProfiles	933
Chapter 194: SetStore	934
Chapter 195: SetStoreCategories	935
Chapter 196: SetStoreCustomPage	936
Chapter 197: SetStorePreferences	937
Chapter 198: SetTaxTable	938
Chapter 199: SetUserNotes	939
Chapter 200: SetUserPreferences	940
Chapter 201: UploadSiteHostedPictures	941
Chapter 202: ValidateChallengeInput	942
Chapter 203: ValidateTestUserRegistration	943
Chapter 204: VerifyAddFixedPriceItem	944
Chapter 205: VerifyAddItem	945
Chapter 206: VerifyAddSecondChanceItem	946
Chapter 207: VerifyRelistItem	947
Chapter 208: VeROReportItems	948
VeROReportItems Input Fields	948
VeROReportItems Output Fields	948
VeROReportItems Details	949
Prerequisites	949
VeROReportItems Best Practices	949
Managing the Volume of the Response	949

Testing VeROReportItems	949
VeROReportItems Sample	949
Related Calls	951
Change History	951
Part 2: Appendices	952
Appendix A: Useful Third-Party Resources	953
Web Services	953
Web Services Projects and References	954
Web Services Specifications and Standards	955
Web Services Journals	956
Base64 Decoding	956
HTML Forms	957
Internet Explorer WebBrowser Control	957
Regular Expressions	957
Other Resources	958
Appendix B: Data Types	959
Summary of Data Types	959
eBay Namespace	959
eBay Types	959
XML Schema Simple Types	959
Time Values	960
Why Do Time Values Seem Inconsistent with the Web Site?	960
About GMT	960
Converting Between UTC/GMT and Local Time	960
Converting Between Pacific Time (PST/PDT) and a Local Time	962
Appendix C: Compatible Application Check for Specific Calls	963
Appendix D: Current Compatibility Levels	965
Appendix E: Error Message Strings	966
Schema Reference Documentation	967
Part 3: Index	968
Index	969

Copyright Notice

Copyright © 2004–2013 eBay Inc. All rights reserved.

This documentation and the API may only be used in accordance with the eBay Developers Program and API License Agreement.

Release Notes

The unified schema API Release Notes for this version of the API are available here:

<http://developer.ebay.com/DevZone/XML/docs/WebHelp/ReleaseNotes.html>

Welcome

Welcome to the eBay Developers Program! The eBay Developers Program provides technology and resources that enable you to fully leverage eBay's platform, strategies and community. The eBay Developers Program promotes your innovation and ideas by providing you with these benefits:

- Our unique marketplace of over 95 million registered users, passionate about buying and selling
- A scalable architecture that has proven itself to be the one of the most available and high-volume commerce engines on the Web
- A large software development team that continues to improve our ecommerce functionality

The Trading API contains calls that return data in a standard format for easy use within your application. Using the Trading API, you can build highly reliable tools and bring them to market faster than is possible with other means.

For a more detailed introduction to the Trading API, please see ["Product Overview"](#) on page 45.

This chapter provides information about resources that are available to help you be a successful eBay developer and a summary of what you will find in this documentation:

- ["Feedback and Support"](#) on page 35
["About the Documentation"](#) on page 39

Feedback and Support

A number of support options are available to eBay Developers Program members. See the following sections for the feedback and support options that eBay offers:

- ["Free Technical and Business Resources"](#) on page 35
["Developer Technical Support"](#) on page 38
["Other Contacts and Resources"](#) on page 39

Free Technical and Business Resources

A number of free resources are available to help you resolve technical issues and to find out about the latest features that are available. See these topics below for details:

- ["eBay Developers Program News"](#) on page 36
["Developer Forums"](#) on page 36
["Knowledge Base"](#) on page 36
["System Announcements"](#) on page 37

eBay Developers Program News

eBay Developers Program News is a newsletter that we release about every two weeks. eBay Developers Program members often use this as their primary source of information about important changes that they should be aware of, new features that are available via eBay Web Services, and other news.

Go to the Account Information section of the eBay Developers Program Web site's Site Map to subscribe or unsubscribe to the newsletter. If you are not the primary contact for your membership (e.g., you work for the individual who is listed as the primary contact), the primary contact can add you as a Newsletter Contact by editing their membership information.

Site Map

<http://developer.ebay.com/sitemap>

Account Information

In your account settings, you can modify various preferences (including newsletter subscription preferences) by modifying your membership information. You can access this information through your Account Information page on the eBay Developers Program site (click **Edit Membership Info** on the Account Information page.)

Account Information

<http://developer.ebay.com/devzone/account/>

Developer Forums

The Developer Forums provide an interactive message board for sharing knowledge and questions about the eBay Developers Program products with other developers like yourself.

We've created these boards to help foster interaction between our developer community and encourage you to:

- Discuss technical issues with other developers like yourself
- Seek brief technical or business-related advice from eBay staff
- Provide eBay with feedback about the eBay Web Services products and documentation

The Developer Forums are accessible from the **Get Help** page on the eBay Developers Program site.

Developer Forums

<http://developer.ebay.com/forums>

Knowledge Base

The Knowledge Base contains tech notes and Q&As that represent common questions we get from developers.

The Knowledge Base is accessible from the **Get Help** page on the eBay Developers Program site.

Knowledge Base

<http://developer.ebay.com/DevZone/support/knowledgebase.asp>

System Announcements

Use our System Announcements forum as your first resource for checking the status of issues relating to the Sandbox and the Production servers.

Note: The Sandbox is a test environment that we make available to you. The Production site is the main eBay site. See “[eBay Sites and Environments](#)” on page 149 for details.

Please check the System Announcements page before reporting any urgent production system issues to eBay. Typical issues we announce on this page are latency, unavailability, known errors, and known call failures.

eBay Developers Program System Announcements Forum
<http://developer.ebay.com/support/system-announcements/>

Site-wide issues that could affect our Production servers are reported on the eBay System Status Announcement Boards.

eBay System Announcements Board
<http://www2.ebay.com/aw/announce.shtml>

Release Notes and Known Issues

In addition to using the technical resources above, be sure to review the latest list of known issues if you encounter unexpected functionality (e.g., something that worked before no longer works) or if you have trouble using the Sandbox.

Note: The Sandbox is a test environment. The Production site is the main eBay site. See “[eBay Sites and Environments](#)” on page 149 for details.

It is also a good idea to check the release notes periodically, as we may make changes that directly or indirectly affect your application.

For the latest release notes, please refer to the eBay Developer Documentation Center.

eBay Developer Documentation Center
<http://developer.ebay.com/support/docs/>

For known issues, please refer to the System Announcements and Knowledge Base.

Links to Site Status Page for Production, the Sandbox, and the SDKs
<http://developer.ebay.com/support/system-announcements/>

Knowledge Base
<http://developer.ebay.com/DevZone/support/KnowledgeBase.asp>

Learning How to Sell on eBay

In addition to this documentation, a variety of resources are available on the eBay Web site to enhance your understanding of eBay features and the basic rules for using the site.

All the business rules that you follow when you use the eBay Web site must also be followed when you access eBay via the Trading API.

eBay offers a number of educational resources to help you make the most of the solutions eBay offers. These include the following topics:

Selling Overview

<http://pages.ebay.com/education/sellingshopping/index.html?ssPageName=MOPS:selltips>

Information for Power Sellers

<http://pages.ebay.com/services/buyandsell/powersellers.html>

eBay Educational Events

<http://pages.ebay.com/university/index.html>

Seller Services

<http://pages.ebay.com/services/sellerservices.html>

The eBay Help Center

<http://pages.ebay.com/help/sell/topics.html>

Developer Technical Support

The options for technical support are:

- Filing a support request
- Live Help

Support Requests and Live Help

<http://developer.ebay.com/support/developersupport/>

The information below should be sent to Developer Support to better enable them to assist you. Some of this information is required, as noted below.

Table F-1 Information to Include When Contacting Support

Information	Required	Description
Call name	Yes	Name of the call that was used when the problem occurred.
Date and time of problem	Yes	Self-explanatory, the date and time (in GMT if possible) that the problem occurred.
Developer ID (DevID)	Yes	This is the developer ID. See “ Security ” on page 101.
Application ID (AppID)	Yes	This is the application ID.
Description of problem	Yes	A concise description of the circumstances surrounding the problem. Be sure to impart all factors that clearly or reasonably might be associated with the problem. For example, does the problem occur consistently or only intermittently? Were peripheral or environmental factors a potential cause of the problem? What steps led up to the problem? How did the problem manifest itself? A clear and complete description of the problem will help Developer Support find and resolve the technical problem for you quickly.

Table F-1 Information to Include When Contacting Support (Continued)

Information	Required	Description
Input data	No	For the API call involved in the issue, provide the input and output XML. When optional fields are involved, be sure to list what optional fields were and were not used. Fields common to all calls (see “ Standard Input Data ” on page 92) should also be mentioned.
Result	No	Describe the result of the failed call. Was there an error? If so, what was the resulting error code and message? Was the error from the eBay Trading API, from a browser, or from the application? See “ Error Handling ” on page 107.
eBay user ID of requestor	No	Provide the authentication information (for example, the user token) associated with the request.
Development language	No	Provide your programming language (for example, Java or C#).

Other Contacts and Resources

For non-technical issues, please refer to the following resources:

Table F-2 Other Points of Contact

Developer Relations	http://developer.ebay.com/contactus/ Contact Developer Relations for help with business issues such as: <ul style="list-style-type: none"> ■ Licensing agreements and uses of the API ■ Conference events, Web-based seminars, and forums ■ Upcoming product changes ■ Getting started with the program, using the Sandbox, retrieving security keys or passwords
API Billing	API-Billing@eBay.com Contact API Billing for: <ul style="list-style-type: none"> ■ Concerns over billing policies ■ Current or overdue invoices
eBay Developers Program site	http://developer.ebay.com

About the Documentation

This manual is available as a PDF and as HTML-based online help.

You can browse and search the HTML version of this documentation on the Web. A link to the online version is available at the Developer Documentation Center on the eBay Developers Program site. You can also download the latest PDF version of this documentation to browse and search it locally.

eBay Developer Documentation Center
<http://developer.ebay.com/support/docs/>

This documentation is updated for each release (see “[The eBay Developers Program Release Cycle](#)” on page 85). The release number appears in the release notes and at the end of each topic (HTML page) in the online help.

Intended Audience

We assume you are an eBay Developers Program member who is comfortable working with XML. This guide doesn't include general information about using XML.

Please see “[Useful Third-Party Resources](#)” on page 953 for additional information.

How this Documentation is Organized

This documentation is organized into several parts.

Introduction to eBay APIs and features

The following parts help you come up to speed on the basics of how to use the Trading API and how to interact with eBay:

- Part I: “[Getting Started](#)” on page 44 gives you an introduction to the Trading API, the use cases it supports, and the types of applications that you can build with it. It describes the basics of how to make a call, and provides basic information on developing, testing, and authentication & authorization. This part also provides a tutorial that helps you make your first call.
- Part II: “[eBay Features Overview](#)” on page 132 introduces important eBay concepts (e.g., items) for developers who may be new to the eBay marketplace.

Core topics

The following parts describe the core information you need in order to build eBay-enabled applications:

- Part III: “[Selling Items on eBay](#)” on page 157
- Part IV: “[Managing Listed Items](#)” on page 374
- Part V: “[Item Specifics and Catalog Products](#)” on page 480
- Part VI: “[Searching for and Retrieving Items](#)” on page 497
- Part VII: “[Retrieving User Data](#)” on page 512
- Part VIII: “[eBay Stores](#)” on page 533

Reference Information

The following parts include details about

- Part IX: “[Specialty and International Sites](#)” on page 556 notes the unique rules and behaviors for eBay Motors and international sites.
- Part X: “[Platform Notifications Guide](#)” on page 620 describes the available notifications and how to use them.
- Part XI: “[API Call Reference Guide](#)” on page 765 focuses on the specifics of how to use the calls that are available in the Trading API . See also “[eBay Trading API Call Reference](#)” on page 41.
- Part 2: “[Appendices](#)” on page 952 contains a number of useful tables and lists for your reference, including special codes that cannot be accessed programmatically (for example, region codes and site IDs), Compatible Application Check Requirements for individual calls, and other useful information.
- Part 3: “[Index](#)” on page 968 contains an alphabetical index to this documentation.

Related Documentation

In addition to the information in this guide, note the following documentation resources for the Trading API.

eBay Trading API Release Notes

The *eBay Trading API Release Notes* list the schema changes for a given release and describe any new features or changed functionality. Check the release notes for important announcements and documentation corrections or enhancements.

[eBay Trading API Release Notes](#)

eBay Trading API Call Reference

Detailed information about each property in the Trading API is provided in the *eBay Trading API Call Reference*. This reference document describes the standard and call-specific input and output fields for Trading API calls.

[eBay Trading API Call Reference](#)

The *eBay Trading API Call Reference* also contains a Type Index page that contains links to all complex and enumerated data type definitions. In addition to defining the type, each Type HTML page contains all fields (with descriptions) of the type, as well as all calls and parent types that use that type. The Type Index can be useful if you are using an eBay SDK and you want to map the types in the schema to classes in the SDK. However, note that some calls only use a subset of the fields defined on a type. (That's why you usually need to look at the “Call” view of the schema.).

[Type Reference](#)

Error Message Strings: The Error Message Strings documentation lists all of the error messages that could be encountered when invoking eBay Web services. The information for each error includes: error code, error severity, short error string, and long error string. All error messages are supplied

only in English, but the error codes can be used as an index for a structure, array, or string table to supply messages in whatever language is desired. If you are reading the PDF version of this guide, you can access the Error Message Strings documentation via the online documentation on the eBay Developers Program site.

Error Message Strings

Some codes that identify listing types appear in the response when you retrieve a listing, but you cannot specify them in requests. See “[Code Lists](#)” on page 82 for information about the notation we use to identify these cases.

When the rules for a given property are not the same for all calls, the rules are described in the documentation.

Typographic Conventions

In this guide, we use the following typographic conventions:

Table F-3 Typographic Conventions Used in this Documentation

Format	Meaning
bold teal	Language- and protocol-independent references to call names and data fields; for example: “Use AddItem to list an item on an eBay site.” Note: In order to perform a particular task on eBay (e.g., listing an item), you need to use a class, module, or function that creates your request, sends it to the eBay Platform, and handles the response. Regardless of how the class is implemented on the client side, it needs to make the call using a particular name that is defined by eBay. The canonical call names are fairly self-explanatory, like AddItem – which you use to list an item.
Dot Separator	Association or containment relationship; for example, Item.ShippingDetails means we are referring to the ShippingDetails field of the Item object.
red text	Deprecated call names and data fields; for example: “ Deprecated: The Foo field is deprecated as of release 751. ” The red color is only a decoration. It is used in combination with a text explanation or a label (e.g., Deprecated) so that the meaning is not dependent on the color. Warnings and alerts may also use red.
pale plum text	Obsolete construct or functionality; for example: “ Obsolete: The Bar field is obsolete as of release 551. ” The plum color is only a decoration. It is used in combination with a text explanation or a label (e.g., Obsolete) so that the meaning is not dependent on the color.
green text	Notes. The green color is only a decoration. It is used in combination with a label (e.g., Note) so that the meaning is not dependent on the color.
blue text	Hyperlinks, including cross references and URLs; for example: “See “ Typographic Conventions ” on page 42.”
monospace	Names of elements, messages, and classes, as well as snippets of sample code; for example: “Use AddItem to list an item on the eBay site.”
italics	New terms; for example: “The <i>Store Inventory</i> format lets sellers list items with longer durations and lower insertion fees.” Placeholder values (variables); for example: “The naming convention is .”

Diagram Conventions

This guide uses the following diagram conventions to convey information about the schema:

Table F-4 Diagram Conventions

Sample Illustration	Meaning
GetItemResponse +	A box with a solid border and a plus sign (+) on one side represents a single element that is required (<code>minOccurs="1" maxOccurs="1"</code>) and that contains child elements.
CategoryType +	A rounded box with a plus sign (+) on one side represents a complex type (which consists of child elements).
PrimaryCategory	An element connected to a yellow box that contains other elements represents a single element (<code>PrimaryCategory</code>) with a reference to an external complex type (<code>CategoryType</code>). The name of the complex type appears at the top of the yellow box that surrounds the complex type's contained elements.
CategoryID	A box with a solid border and no plus sign represents single element that is required (<code>minOccurs="1" maxOccurs="1"</code>) and that contains Parsed Character Data (#PC-Data). The content may be simple content or mixed complex content.
AutoPayEnabled	A box with a dotted border indicates a single element that is not necessarily required (<code>minOccurs="0" maxOccurs="1"</code>). Note: An element that is defined as <code>minOccurs="0"</code> may be required for some calls. See "Base Components" on page 79 for more information.
CSIDList	A layered box with a dotted border represents a multiple element that is not necessarily required (<code>minOccurs="0" maxOccurs="unbounded"</code>). Note: An element that is defined as <code>minOccurs="0"</code> may be required for some calls. See "Base Components" on page 79 for more information.
AbstractRequestType	Solid lines connect parent elements (<code>AbstractRequestType</code>) to required child elements (<code>Version</code>). Dotted lines connect parent elements to child elements that are not necessarily required (<code>ErrorLanguage</code>).

Part I

Getting Started

Product Overview	45
System Requirements	50
Executing Your First API Call	55
Developing an eBay-Enabled Application	56
Invoking eBay Trading Web Services	74
Getting Tokens	117

Chapter 1

Product Overview

The eBay Developers Program enables third-party developers to create innovative solutions that tap into the power of eBay, The World's Online Marketplace®.

Many users buy and sell items using the eBay Web site, interacting with eBay directly through our standard user interface. As a member of the eBay Developers Program, you have access to eBay's platform and technical resources. eBay Web Services offers application developers the power to create a wide variety of solutions to meet the diverse needs of the more than 135 million buyers and sellers on eBay. By leveraging the resources offered by the eBay Developers Program, you can create your own applications that communicate directly with the eBay business logic and database. Your applications can provide a custom interface, functionality, and specialized operations that are not available in the eBay Web site user interface.

This chapter covers:

- “[eBay Trading Web Services](#)” on page 45
- “[eBay SDKs](#)” on page 46
- “[Supported Use Cases](#)” on page 47
- “[Types of Applications You Can Build](#)” on page 48
- “[API License Agreement and Usage](#)” on page 49

eBay Trading Web Services

eBay Web Services provides programmatic access to the eBay marketplace. It enables third-party applications to build custom applications, tools, and services that leverage the eBay marketplace in new ways.

You can select a programming language (such as PHP, .NET, Java or any other language that you are comfortable with) to access eBay Web Services. We also have software developer kits (SDKs) for Java and .NET to make your programming tasks in these languages easier. Go to the eBay SDKs page to learn about and install the eBay SDKs.

eBay SDKs
<https://www.x.com/developers/ebay/documentation-tools/sdk>

Typical eBay-enabled applications include item sales management (which could include integration with database inventory systems), item search utilities, user account management systems, systems that embed partner item listings in their Web pages, and applications that assist buyers in managing the listings they're interested in.

The power of eBay Web Services lies in making these applications possible independently of the eBay user interface. That is, your application doesn't need to change every time the eBay user

interface changes, and it doesn't need to present data in the same way eBay does. Your eBay-enabled application can present data in custom ways that best meet your users' needs.

eBay Web Services supports two protocols:

XML/HTTPs: With the XML API, the request interface is an XML document. Each request is composed of XML elements that specify the request parameters (e.g., the properties of an item to be listed and additional processing instructions).

With the XML API, your application builds the request as an XML string, sets a number of HTTP headers, and sends the request to eBay using the HTTPS protocol. After processing the request, the eBay server sends a response back to your application, also via HTTPS. This response consists of an XML document containing the data that resulted from the original request. Your application then parses the XML to extract the data.

SOAP: With the SOAP API, the request interface is an object in your application's native programming language. You use a third-party SOAP client to generate business-object interfaces and network stubs from a WSDL document that specifies the eBay message schema, the service address (eBay's SOAP gateway URL), and other information. Your application works with data in the form of object properties, and it sends and receives the data by calling object methods. Your SOAP client handles the details of building the SOAP request and sending it to eBay, and converting the response back to an object that is easy to work with. This frees you from the need to build and parse XML documents yourself, so you can focus on managing and presenting the data itself. By simplifying the way you access eBay data, the SOAP API helps you to get your applications up and running more quickly and to adapt more easily to changes.

The SOAP API is built on open standards like SOAP and WSDL. These standards are supported by a wide-range of development tools on a variety of platforms.

Both the XML API and the SOAP API share the same schema, so the basic format of the input and output data is the same no matter which API you use. Regardless of whether you are using the SOAP API or the XML API, you are accessing the same eBay functionality and data. Thus, you can choose to use one or the other or both—whichever approach is best for you.

eBay SDKs

For the Trading API, there are two SDKs (Software Development Kits) that are designed to make application development easier and more efficient. SDKs are downloadable kits that include pre-coded solutions for common programming tasks in a language you already use.

SDKs help simplify some common programming tasks, such as error handling and call retry. See below for the SDKs that apply to the Trading API. SDKs are based on a specific API version.

eBay SDK for Java: The eBay SDK for Java is designed for use in creating applications in various environments (such as Microsoft Windows and Linux) using Java. The eBay SDK for Java is an encapsulation of the SOAP API. Developers can use the eBay SDK for Java to create applications that

integrate with eBay, including classes for automating data synchronization. The eBay SDK for Java supports programming in the Java language.

eBay Software Development Kit for Java
<https://www.x.com/developers/ebay/documentation-tools/sdks/java>

eBay SDK for .NET: Designed for the Microsoft Windows environment, the eBay SDK for .NET is a set of class libraries that encapsulate the SOAP API. Developers can use the eBay SDK for .NET to create applications that integrate with eBay, including classes for automating data synchronization. The eBay SDK for .NET supports programming in any .NET language or language that can work with COM, including C#, VB.NET, and VB 6.

eBay Software Development Kit for .NET
<https://www.x.com/developers/ebay/documentation-tools/sdks/dotnet>

Note for SDK Users: The fields in this documentation are defined in the WSDL upon which the SDK is based – the field names are not based on the corresponding SDK wrapper class. Because of this, there can be instances where the field names in this reference do not match the names in the SDK wrapper classes. However, the behavior of the corresponding fields is the same.

This documentation applies to a specific schema version, and it might contain fields that are unavailable in the version of the SDK you're using. Advanced SDK users can use such fields by updating the code stubs in their SDK (using a more recent WSDL file than the one delivered with the SDK). For more information on WSDL updates (or code generation), see the Readme file for the SDK you are using.

This documentation shows the language-independent simple types (e.g., `string`). For each field, you can find an equivalent class, method, property, or variable in the SDK. Start by looking at the methods and properties that are defined on the wrapper class.

Supported Use Cases

You can use the API to perform these tasks, and more:

- Retrieve the eBay category hierarchy (see “[Retrieving the Category Hierarchy for a Site](#)” on page 158)
- Submit a listing to eBay (see “[Listing Items](#)” on page 174)
- Verify the validity of the data in your listing before you actually submit it to eBay (see “[Listing Items](#)” on page 174)
- Resubmit a listing that did not end in a purchase (see “[Relisting Items](#)” on page 376)
- Retrieve multiple listings for a single seller (see “[Browsing a Seller’s Items](#)” on page 502)

- Retrieve information about a single listing (see “[Retrieving the Data for One Item](#)” on page 504)
- Retrieve all the listing-related events (e.g., the fact that a listing ended) for a single seller (see “[Working with Seller Events](#)” on page 391)
- Retrieve single line item orders (see “[Retrieving the Order Line Items for a Single Item](#)” on page 418)
- Retrieve orders for a particular seller (see “[Retrieving the Order Line Items for a Specific Seller](#)” on page 419)
- Get information about a single user (see “[Retrieving User Information](#)” on page 513)
- Search for items and retrieve detailed information about items found (see “[Searching for Items](#)” on page 498)

Types of Applications You Can Build

This section focuses on some of the types of applications, tools, and services that can be created using the API.

This list is just a sampling of what is possible. The variations are many. Perhaps you came to the eBay Developers Program with a specific business need to be satisfied by an application based on eBay technology. You might be a developer looking to break into eCommerce application development, without a specific type of application in mind. Use this section to help guide you in creating the application you already have in mind or to spur your imagination with a type of application you never even imagined before.

- “[Listing Application](#)” on page 48
- “[Seller Application](#)” on page 49
- “[Gallery List Tool](#)” on page 49

Listing Application

A listing application is a relatively simple program that provides a seller with the ability to list items on eBay. The breadth of coverage for such an application is fairly focused and so the technology required to bring the application to existence is similarly narrow.

If the user already knows the Category ID of the eBay category in which they want to list their item(s), the minimum requirement for a basic listing application is that it should support submitting a listing to eBay (see “[Listing Items](#)” on page 174). A user might already know the eBay Category ID if they sell in a vertical market.

If the user does not know the eBay Category ID, the basic listing application should also support the ability to retrieve and display the eBay category hierarchy and allow the user to select a category (see “[Retrieving the Category Hierarchy for a Site](#)” on page 158).

Seller Application

A seller application is a more comprehensive solution for sellers than a listing application. A seller application covers most, if not all, of the aspects involved in the sales process, from beginning to end. This includes:

- Listing items (see “[Listing Items](#)” on page 174)
- Monitoring buyer bidding (see “[Browsing a Seller’s Items](#)” on page 502)
- Discovering current bidders in an auction listing or buyers of fixed-price items (see “[Browsing a Seller’s Items](#)” on page 502)
- Processing and fulfilling items that have sold (see “[Retrieving the Order Line Items for a Single Item](#)” on page 418 and “[Retrieving the Order Line Items for a Specific Seller](#)” on page 419)

Gallery List Tool

A gallery list tool is one that displays a seller’s items in a sub-section of a Web page that has other content. For example, a Web page with information about professional golfing might have a rectangular area on the page in which the page’s host offers golf equipment for sale. The tool would retrieve a collection of items the seller currently has up for sale on eBay.

See “[Browsing a Seller’s Items](#)” on page 502 for information about functionality that would support the development of a gallery list tool.

API License Agreement and Usage

Your use of the API is subject to all the rules described in your API Public License Agreement. The same Compatible Application Check Requirements also apply regardless of the protocol you use to access the eBay Platform.

Your API Public License Agreement
<https://developer.ebay.com/join/licenses/individual/api.pdf>

Chapter 2

System Requirements

This chapter describes the operating system, hardware, and database requirements for using the API.

- “[Supported Client Environments](#)” on page 50
- “[Where to Find the eBay Schema Files](#)” on page 51
- “[Database Storage](#)” on page 53
- “[Hardware Requirements](#)” on page 53
- “[Next Steps](#)” on page 54

Supported Client Environments

The API has been tested with the following client environments.

Table 2-1 Supported Client Environments

Client	Programming Environment	Operating Environment
Microsoft .NET 1.1 Framework	Microsoft Visual Studio .NET	Microsoft Windows 2000 and Windows XP
Apache Axis 1.1 Final	Java (J2SE 1.3.1 and 1.4.X)	Linux and Microsoft Windows 2000

For information about programming using the Java or .NET eBay SDKs, see “[eBay SDKs](#)” on page 46.

If you choose to try a different client, please be sure it supports document-style messaging (as we do not use RPC-style messaging). Also, the API only supports the UTF-8 encoding scheme.

For information and downloads, please see the following sites.

For Java development:

Sun Microsystems, Inc.
<http://www.oracle.com/technetwork/indexes/downloads/index.html>

The Apache Software Foundation
<http://axis.apache.org/axis2/java/core/>

Note: If you use Apache Axis, you only need the .jar files that are necessary for consuming Web services (i.e., on the client side). However, it may be convenient to install the product on an application server so that you can test your ability to post SOAP messages to a server other than eBay’s SOAP API gateway. The standard installation instructions for Apache Axis assume you are installing the product on an application server and include information about where to find a free application server.

For .NET development:

Microsoft .NET Framework Versions 1.0 and 1.1

Microsoft Corporation

<http://www.microsoft.com/downloads/details.aspx?FamilyID=262d25e3-f589-4842-8157-034d1e7cf3a3&DisplayLang=en>

Where to Find the eBay Schema Files

The XML API and the SOAP API are based on the same schema. If you are using the XML API, you can use ebaySvc.xsd for request validation and in conjunction with some XML tools that recognize the format. If you are using the SOAP API, you will need to generate eBay API interfaces and proxies from ebaySvc.wsdl.

[“URLs for the eBay Schema Files” on page 51](#)

[“Generating Interfaces and Proxies from a WSDL” on page 52](#)

URLs for the eBay Schema Files

You can access a particular version of the eBay schema using an URL with the following format, where **VERSION** is the version identifier of the release:

ebaySvc.xsd

<http://developer.ebay.com/webservices/VERSION/ebaySvc.xsd>

ebaySvc.wsdl

<http://developer.ebay.com/webservices/VERSION/ebaySvc.wsdl>

The version identifier can be:

- A number (for example, 813) – This is the version number of a particular schema file (a release number).
- **latest** – This always points to the schema with the highest version number.

For example, you can access version 813 of the schema here:

Version 813of ebaySvc.xsd

<http://developer.ebay.com/webservices/813/ebaySvc.xsd>

Version 813 of ebaySvc.wsdl

<http://developer.ebay.com/webservices/813/ebaySvc.wsdl>

If 813 were the latest version, it would be available here:

Latest ebaySvc.xsd
<http://developer.ebay.com/webservices/latest/ebaySvc.xsd>

Latest ebaySvc.wsdl
<http://developer.ebay.com/webservices/latest/ebaySvc.wsdl>

All releases have odd-numbered versions. Older versions are periodically deprecated, which means they are no longer supported. See the [eBay Trading API Schema Versioning Strategy](#) for details about how deprecation works for the Trading API and a list of supported versions.

Each time we release a new version of the eBay schema, we add a new directory with a new version number, and point the “latest” URL to the new version. That is, the schema file in the “latest” directory changes for every release, and the schema files in the numbered directories do not change.

If you are using the SOAP API and you do not plan to update your application each time a new version of the WSDL becomes available, you should use the URL with the numbered version identifier. If you plan to always retrieve the latest changes available, it may be more convenient to use the URL with the “latest” version identifier.

See “[Release Versions](#)” on page 84 and, for the SOAP API, see “[When to Update the Schema](#)” on page 86 for additional information about versioning. If you use an IDE that recognizes WSDL files and displays the annotations, you may find it useful to browse the eBay WSDL file. You can also find the type annotations in the Type Reference (see “[How this Documentation is Organized](#)” on page 40).

Generating Interfaces and Proxies from a WSDL

If you are using the SOAP API, please refer to your SOAP client documentation for information about how to generate eBay API interfaces and proxies from the eBay WSDL.

[Example 2-1](#) shows a simple way to generate Java source files from the eBay WSDL (using a Windows command prompt). In this example, the command-line arguments are wrapped to fit the width of the page. Because the eBay WSDL is so large, the timeout for the command is disabled with **-O -1** (**Note:** the letter O, not the number 0). In practice, the information would all be entered on one line.

Example 2-1 Generating Interfaces and Stubs from the eBay WSDL (Java)

```
java org.apache.axis.wsdl.WSDL2Java -o src -O -1
http://developer.ebay.com/webservices/813/ebaySvc.wsdl
```

If you are using Microsoft Visual Studio.NET, follow the application’s instructions to add a Web Reference to your solution. Specify the eBaySvc.wsdl URL as the Web service URL.

URL for the API Gateway

All API requests that you execute must be sent to eBay’s Sandbox or Production API Gateway. See “[Routing the Request \(Gateway URLs\)](#)” on page 87 for details about the API Gateway URLs and how to configure them.

Database Storage

Using the API, data is submitted to or retrieved from eBay using a single call. This data is transient, existing only as long as the application happens to be running. For the data to be of most use, it should be retained by the application from one session to the next. For example, to work with the data for a particular item, an application could retrieve that data each time the application needs that data. The better approach, though, would be to retain the data and then refer to it as needed.

One of the best approaches for retaining data in an application is to store it in a database. Database systems afford built-in means for easily querying, modifying, and sorting data.

The API imposes no requirements or limitations on which particular database system an application can or should use to manage its data. Basic qualifications for database systems and where the database system would be accessed from (Web versus local machine versus network server) should drive this decision. See the specifications provided by individual database system vendors to ensure that they meet the particular needs of your application. Not all database systems can be used with Web-based applications.

Commonly Used Database Systems

There are a number of commercial database systems that can be used in an application, both for local databases and Web-based systems. Here are a few.

Access
Microsoft Corporation
<http://www.microsoft.com/office/access/default.asp>

SQL Server
Microsoft Corporation
<http://www.microsoft.com/sql/default.asp>

Oracle: Web Services Center
Oracle Corporation
<http://www.oracle.com/us/products/database/index.html>

MySQL Products
MySQL
<http://www.mysql.com/products/index.html>

PostgreSQL
<http://www.postgresql.org/>

Hardware Requirements

If an application is to be a stand-alone executable that the end-user installs on a local machine, then hardware requirements will be the responsibility of the end-user. Naturally, the application developer should provide some guidelines in this regard: recommended minimum system

requirements. However, today most computer systems are very powerful and relatively inexpensive, and all should have the memory, CPU speed, and disk space needed for nearly all eBay-enabled applications.

But if the application is to be Web-based, the developer or hosting company must have available and maintain the hardware from which the application would be hosted. (In addition to the hardware itself, this also requires the Web-hosting capabilities inherent in providing access to it from the Internet.)

Some general things to consider when making hardware decisions for an application:

- You can rent the hardware on which an online application is hosted from an Internet Service Provider (ISP).
- It is very useful to have administrator control of the server on which the application is hosted.
- Not owning the hardware (i.e., it is rented from an ISP) sometimes makes diagnosing and debugging application problems more difficult.

Next Steps

When you are ready to start using the API, it is suggested that you review the following topics:

- If you are new to the eBay Developers Program, see “[Product Overview](#)” on page 45.
- If you are new to using eBay, see “[eBay Concepts](#)” on page 133.
- If you want to start learning what you need to develop and test an eBay-enabled application, see “[Developing an eBay-Enabled Application](#)” on page 56.
- If you want to learn how to send data to eBay, see “[Invoking eBay Trading Web Services](#)” on page 74.
- If you are not already familiar with the various help and documentation resources that are available to you, see “[Feedback and Support](#)” on page 35 and “[About the Documentation](#)” on page 39.

These sections contain useful links to technical support, free help resources, and other documentation, including the error message string documentation.

Chapter 3

Executing Your First API Call

Refer to the Trading API Tutorials page (<http://developer.ebay.com/DevZone/XML/docs/HowTo/index.html>) in the eBay Trading API documentation for descriptions and links to tutorials. The tutorials provide instructions and sample code for making your first eBay API call. Tutorials are provided for a variety of languages and protocols.

Chapter 4

Developing an eBay-Enabled Application

This chapter provides some advice on how to plan ahead. Several practices for your application are recommended. This chapter also describes resources for helping you test your application and describes important requirements your application needs to adhere to before we will allow you to send real data to the eBay Web site.

An eBay-enabled application that is sending real data to the eBay Web site is sending calls to the *Production environment* (also called the *Production site*). A user who views the eBay Web site through a browser is viewing that data in the Production environment.

During the development of your application, you will probably simulate various tasks related to buying and selling on eBay. The test environment is called the *Sandbox*. You can create one or more users and set them up as sellers to simulate listing items to the Sandbox. You can create users to act as buyers to bid on items via the Sandbox (see “[PlaceOffer](#)” on page 904). You can even expand your own test suites by combining your efforts with those of other members of the eBay Developers Program and testing each other’s listings. See “[The Sandbox and Production Environments](#)” on page 149 for important information about working in these environments.

Note: The Sandbox may be unavailable due to maintenance on Wednesdays from 8pm-10pm Pacific Time.

In the Sandbox, you can make as many calls as you need. However, in the Production environment, your application needs to meet certain requirements. Please be aware of the requirements and the process in advance and design your application accordingly: see “[Planning Ahead](#)” on page 56 and “[Checklist for Going Live](#)” on page 67.

To access the Sandbox, you need a set of Sandbox Keys. These are a set of IDs that are issued to you when you first join the eBay Developers Program. The Sandbox Keys cannot be used to access the Production site.

- “[Planning Ahead](#)” on page 56
- “[Following Best Practices](#)” on page 58
- “[Selecting Fields to Retrieve](#)” on page 59
- “[Testing Applications](#)” on page 61
- “[Checklist for Going Live](#)” on page 67
- “[Sandbox and Production Keys](#)” on page 71

Planning Ahead

Some developers might want to experiment with calls to get their bearings and understand how to use the eBay API. Others might dive right in and begin designing their application, learning as they go. How you choose to approach your design and implementation is up to you. However, you may be able to save yourself some time (and money) later on if you familiarize yourself with the eBay Developers Program resources and Production requirements before you begin development. Use the

information in this guide and the features, resources, and guidelines that are available to you both on the eBay Developers Program site and the main eBay site. For example:

- **What are the guidelines for using the eBay API in the Production environment?**

Your understanding of the API usage rules and the deployment process will affect how you design your application's business logic and your project schedule. For example, your application must pass the Compatible Application Check before going live. You should be aware of the Compatible Application Check timeline, so that you make sure you allow sufficient time for the Compatible Application Check process in your development and deployment schedule. See "[Planning for the Compatible Application Check](#)" on page 57 and "[Compatible Application Check](#)" on page 67.

- **How should you manage data storage?**

Please see "[Database Storage](#)" on page 53 for suggestions.

- **What eBay features does the eBay API support and to what extent are they supported?**

Perhaps you came to the eBay Developers Program with a specific business need to be satisfied by an application based on eBay technology. You might be a developer looking to break into eCommerce application development, without a specific type of application in mind. Before you begin development, it is a good idea to browse this documentation to gain a clear understanding of the features that are available in the eBay API.

See "[Supported Use Cases](#)" on page 47 and "[Types of Applications You Can Build](#)" on page 48 for information on just some of the uses of the eBay API.

- **How can you improve your selling strategy and what services might make your seller application stand out?**

Whether you're building an application to integrate your own retail services with eBay or building an application for other sellers to use, you can take advantage of the power-selling tips that eBay offers. For example, if your listing ends and your item has not sold, you may want to take advantage of the relisting credit that eBay offers. Tips like this are described in this guide (see "[Managing Item Listings](#)" on page 375). A great deal of information is also available on the eBay Web site. See "[Learning How to Sell on eBay](#)" on page 37 for a list of suggested resources.

Planning for the Compatible Application Check

When an application has passed the Compatible Application Check, it means the application complies with the Compatible Application Check Requirements (see "[Compatible Application Check](#)" on page 67).

You apply for the Compatible Application Check after you have finished developing and testing your application. However, you should plan and design your application with the Compatible Application Check Requirements in mind.

The Compatible Application Check Requirements identify performance criteria, polling frequency limits, and error-handling specifications. For example, you can submit multiple listings simultaneously by using separate threads. This will speed up the listing process. However, as we allocate a certain amount of resources to the eBay API, the maximum number of simultaneous

threads permitted per call varies according to your application type. Be sure to consider this in your design. See “[Requirements for Sending Simultaneous Requests](#)” on page 69.

It’s important to understand these kinds of requirements before you begin development because they will affect decisions you make concerning data caching, the number of times you call a given function, and other aspects of your design. See “[Compatible Application Check for Specific Calls](#)” on page 963 for requirements that are defined for various calls.

The rules outlined in the Compatible Application Check Requirements are not the sole criteria for judging an application. We modify the requirements based on observations of how developers use the eBay API. Our intent is to ensure that your application is designed to communicate with eBay servers in an efficient and appropriate manner.

In the Compatible Application Check, we review and test applications to ensure that they comply with the Compatible Application Check Requirements. This process can take up to seven (7) business days if you design your application to comply with the requirements and you follow the steps of the Compatible Application Check process carefully. The process can take substantially longer if you do not. You may want to build time for this process into your project schedule. See “[Getting a Compatible Application Check](#)” on page 70 for more information about the process.

Following Best Practices

To make efficient use of your eBay API calls, your application should cache data locally and avoid retrieving duplicate data multiple times. As you develop your application, it is a good idea to follow these practices:

- Build flexibility into your application
- Follow the best practices (if any) that eBay defines for each call
- Understand the eBay release cycle
- Handle all errors
- Understand the tags and functionality that trigger additional seller fees

Building Flexibility into Your Application: The eBay Developers Program supports several versions of the API at any given time. This gives developers more time to adopt changes that are made to the eBay Web site. However, the Developers Program does set a minimum level of support for outdated functionality. See our [eBay Schema Versioning Strategy](#) for details.

Following Best Practices for Specific Calls: The API provides several calls that provide similar services. To make an application more efficient, some developers prefer to choose calls that return more data. This will reduce the number of calls you make. For example, when you are checking to see whether any bids have been placed on a seller’s listings, [GetSellerEvents](#) is more efficient than [GetSellerList](#), and both are more efficient than [GetItem](#). Instead of invoking [GetItem](#) multiple times to retrieve information for each of a seller’s listings, you can invoke [GetSellerEvents](#) or [GetSellerList](#) once and retrieve a more sparse set of information about many items the seller has recently sold. See “[Browsing a Seller’s Items](#)” on page 502 and “[Completing the Sale](#)” on page 415 for more information.

Furthermore, you can reduce the number of order retrieval calls you make by subscribing to eBay's Platform Notifications. When a seller's listing ends or when the seller receives feedback, eBay sends you a notification. Notifications are sent via SOAP messages that eBay posts to a URL you specify in advance. See "Completing the Sale" on page 415 for more information.

Certain calls, such as [GetCategories](#), return very large result sets. In some cases, you may need to wait several minutes for the data to be downloaded-particularly during peak hours. If the data includes historical information or information that does not change frequently, we recommend that you store all the data initially, and then use the provided filters to retrieve incremental updates only.

Release Cycles: You should also be aware of eBay's release cycles (see "[The eBay Developers Program Release Cycle](#)" on page 85). While you are not required to upgrade each time, some of the new features may be of interest, and some functionality that you currently use might be affected. Therefore, be sure to take a look at the release notes regularly. The community newsletter (eBay Developer News) also provides important information about updates to the API.

Handling Errors: If your request is processed successfully but it included small problems, the response may include warnings that explain the problem.

If your request cannot be processed successfully for some reason, the response returns errors instead. These errors provide information indicating the cause of the problem and a unique identifier (error code) for the particular error state that occurred. As with all applications, your application should implement at least basic error handling. If you want to handle certain errors in a special way, you can parse the errors to check for particular error codes. See "[Error Handling](#)" on page 107.

However, sometimes no errors are returned but your application will still fail because an expected field is missing in the response. Thus, in addition to checking the response for errors, you should check to make sure it contains the data you expected before you try to use the data.

Understanding When Additional Fees are Charged: When you list an item on the eBay Web site, the Sell Your Item pages identify features that require the seller to pay additional fees. All fees that are charged when a seller lists an item through the eBay Web site are also charged when an application lists an item through the API. For example, if you set a field that specifies a bold title when you list an item (see "[Listing Items](#)" on page 174), the item title will appear in bold on the eBay site and the seller will be charged a fee. We recommend that you design your application to inform users when they will be charged additional fees. See the eBay online help for information about seller fees.

Fees Overview
<http://pages.ebay.com/help/sell/fees.html>

Selecting Fields to Retrieve

You can use the [OutputSelector](#) field to select the data being retrieved by many "Get" calls in the Trading API, like [GetItem](#), [GetSellerEvents](#), or [GetOrders](#).

The [OutputSelector](#) field can make the response of a Trading API call more manageable, especially when the standard call returns a large payload. If you use the [OutputSelector](#) field, the output data will include only the fields that you specified in the request.

An example of an application that could use an **OutputSelector** inclusive filter is a buying application that displays information to users that they need to buy an item. Using **OutputSelector** tags, the application would specify the following fields in a **GetItem** call: **PictureURL**, **UserID**, **BuyItNowPrice**, **EndTime**, and **ViewItemURL**.

Working with the Output Selector

By using **OutputSelector** filters, you select the fields you want returned in the response. If a field is returned, that field's parent and any child fields are also returned. An **OutputSelector** field cannot be used to specify a field attribute, e.g. **currencyID**.

Below is an XML snippet of how you would specify the **OutputSelector** fields in a **GetItem** call to return **ViewItemURL** (the URL where a user can view the listing) and **BuyItNowPrice** (the price that a potential buyer can purchase an auction item enabled with the **Buy It Now** feature):

Example 4-1 Restricting a GetItem Response to ViewItemURL and BuyItNowPrice

```
...  
<ItemID>4046156497</ItemID>  
<OutputSelector>Item.ListingDetails.ViewItemURL</OutputSelector>  
<OutputSelector>Item.BuyItNowPrice</OutputSelector>  
...
```

When you specify multiple fields, you can either use separate, repeating **OutputSelector** tags for each desired field, as above, or one **OutputSelector** tag with multiple field values delimited with a comma, as below.

Example 4-2 Restricting a GetItem Response Using a Comma-Separator

```
...  
<ItemID>4046156497</ItemID>  
<OutputSelector>Item.ListingDetails.ViewItemURL,Item.BuyItNowPrice</OutputSelector>  
...
```

Note that unspecified fields are not returned. For example, even if **PaginationResult** normally would always be in the response, the use of one or more **OutputSelector** fields will prevent it from being returned, unless **PaginationResult** is among the fields specified with the **OutputSelector** field. For more information, see the following Knowledge Base article:

http://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=1220

You do **not** have to specify the full path of the field unless the field you are specifying can be returned in multiple containers. If the field you are specifying can be returned in multiple containers, e.g. **StartTime** in **GetItem**, then specify the full path, as in the following example:

Example 4-3 Using the Full Path to Specify Item.ListingDetails.StartTime in GetItem

```
...  
<ItemID>4046156497</ItemID>  
<OutputSelector>Item.ListingDetails.StartTime</OutputSelector>  
...
```

If you decide to specify the full path, do not specify the call request type (e.g., do not specify `GetItemRequestType` in the `OutputSelector` field).

If you enter a typo in the field name, an error message is returned, but if you enter an unavailable field, an error message is not returned.

An output selector cannot filter out standard fields from a response, e.g. `Timestamp`, `Ack`, `Version`, and `Build`. Additionally, an output selector cannot specify that only one attribute set or attribute be returned.

The `OutputSelector` field is not available for use with the REST API.

The `OutputSelector` field is supported for most calls that retrieve data. If the `OutputSelector` field is supported for a call, the `OutputSelector` field is listed as an input field for the call.

Best Practices for the Output Selector

In API calls, some fields are only returned if a specific detail level or specific “include” element (such as `IncludeExpressRequirements`) is used. If you use an `OutputSelector` field but do not, for example, use a necessary detail level, then the field specified in your output sector will not be returned. See “[Specifying Detail Levels and Granularity Levels](#)” on page 98.

Thus, the `OutputSelector` field cannot work for a field that would not otherwise be in the response. It is recommended that you make a test call without an `OutputSelector`, confirm that the desired field is in the response, and then make the call again using an `OutputSelector` for your desired field.

Note that if you use an `OutputSelector` tag for a field that occurs multiple times in a response, the field is returned for each instance it would have been returned if an `OutputSelector` field had not been specified.

Testing Applications

Before you begin running an application in the Production environment, you should fully test the application’s features and functions to make sure it operates in an error-free manner. In addition, you must be certain that it abides by all API usage rules.

To help with your testing, eBay provides the *Sandbox environment* (or the *Sandbox* for short), a special environment designed just for testing. In the Sandbox environment, an application can perform all the same operations it would in the Production environment, except that all users, items, and payment funds involved in the orders are fictitious.

Test users, items, and funds in the Sandbox environment cannot be seen or used in the Production environment. Conversely, real users, items, and money from the Production environment cannot be seen or used in the Sandbox environment.

This section provides information and tips for setting up test users and testing applications on special sites (such as eBay Stores and eBay Motors). The following sections are covered:

- “Using the Sandbox” on page 62
- “Creating a Test User” on page 63
- “Generating Auth Tokens for Test Users” on page 65
- “Testing Your Application’s API Access” on page 66
- “Creating Stores in the Sandbox” on page 67

For specifics and tips on testing individual calls, refer to documentation on those calls in the *Trading API Call Reference*. For example, for information on testing [GetAccount](#) in the Sandbox, see the following section in the *Call Reference*:

[GetAccount](#)

About eBay Login Credentials

The following login credentials are completely independent and not interoperable across the following sites.

Important: You can avoid common problems that new eBay developers encounter if you understand the differences between the following credentials.

Table 4-1 eBay Login Credentials

Login Credentials	Used By	Site Where Used	Get Credentials
eBay username and password	eBay member (real buyer or seller)	eBay site UI www.ebay.com (and corresponding international sites)	https://signin.ebay.com/ws/eBayISAPI.dll?SignIn (for eBay US site)
eBay Production authentication token	API calls on behalf of a real eBay user	eBay API https://api.ebay.com (programmatic access only)	
eBay Developers Program username and password	Developer	eBay Developer site developer.ebay.com	https://developer.ebay.com/base/membership/signin/default.aspx
eBay Sandbox username and password	Test user (test buyer or test seller)	eBay Sandbox UI sandbox.ebay.com	“Creating a Test User” on page 63
eBay Sandbox authentication token	API calls on behalf of a test user	eBay Sandbox API https://api.sandbox.ebay.com (programmatic access only)	“Generating Auth Tokens for Test Users” on page 65

Using the Sandbox

Before you can operate an application in the Sandbox, you must obtain a set of *keys* for the Sandbox. You can get the keys by registering and logging into the eBay Developers Program at <https://developer.ebay.com/base/membership/signin/default.aspx>.

Operating an application in the Sandbox is essentially the same as operating it in the Production environment. The main difference is that the application needs to be pointed to the Sandbox environment instead of the Production environment. See “[Routing the Request \(Gateway URLs\)](#)” on page 87 for more information on how to point an application to either the Production or the Sandbox environment.

The Sandbox duplicates much of the functionality that is on the eBay Web site. It processes API calls in exactly the same manner as does the Production environment, and it includes a subset of the eBay site’s Web pages. The difference is that all the users are test users (created by developers like you) and all the money exchanged is test money.

Use the Sandbox to experiment with function calls, to prototype new routines, to test your application’s business logic, and to make sure your application adheres to eBay’s Compatible Application Check Requirements before you actually apply for the Compatible Application Check.

Before you begin working with the eBay API, and before you attempt to connect to the Sandbox, make sure all the tools in your local development environment work properly.

The sooner you start using the Sandbox, the sooner you will gain expertise in how to send requests to eBay and how to handle the results that are returned.

Automated Sandbox Test Users

Prior to creating your own test users (see below), you can use several automated Sandbox users (provided by eBay) to facilitate your testing. The automated Sandbox users are active in the Sandbox—they regularly add items, buy items, pay for items, leave feedback, open unpaid item or item not received cases, and engage in other activities. The automated Sandbox users enable you to test many calls, such as [AddItem](#), [GetItem](#), [GetBidderList](#), [GetSellerList](#), [GetItemTransactions](#), [GetOrders](#), and [GetItemShipping](#).

The following table contains the user names and passwords of the automated Sandbox users:

Table 4-2 Automated Sandbox User Names and Passwords

User Name	Password
external_api_seller	123456
external_api_buyer	123456
external_api_buyer2	123456
external_api_buyer3	123456

Make use of these test users in your API calls by obtaining Auth Tokens for them using the process described below in “[Generating Auth Tokens for Test Users](#)” on page 65.

For definitions of the Sandbox and Production environments, see “[The Sandbox and Production Environments](#)” on page 149.

Creating a Test User

Testing an application in the Sandbox requires the use of one or more test users. Test users act as the sellers and buyers of items during the application-testing process in the Sandbox.

Although eBay does provide a few automated Sandbox users (as described above), it is important to create your own test users so you can completely control their Sandbox activities as appropriate for your application's needs.

Registering and Configuring Test Users

The process of registering a test user is different than that for registering a real user in the Production environment. After you register a test user, you can use it to buy and sell test items in the eBay Sandbox.

To create a test user in the Sandbox environment, use the [Sandbox User Registration Tool](#) on the eBay Developers Program site, and follow the instructions to complete the registration process for a new test user.

Here's how to navigate to the Sandbox User Registration tool and register a test user:

- 1 Go to the eBay Sandbox User Registration Tool page:

<https://developer.ebay.com/DevZone/SandboxUser>

- 2 Log in as a developer, using your eBay Developers Program username and password.
- 3 Create your test user.
- 4 Save the test user's username and password to a place where you can access them as needed.

Please note that eBay automatically adds "TESTUSER_" as a prefix to your test username. Make sure you always include that prefix when you log in as the test user.

- 5 If necessary, use the tool to create additional test users.
 - If you are testing *only* the listing of items, you need only one test user.
 - To test both the buying and selling aspects of an application, you'll need to create at least two test users: One test user to act as the seller and one to act as the buyer. This is because eBay rules forbid a seller from bidding on their own listings, a rule that also applies in the Sandbox.
 - To fully test multi-quantity fixed-price listings, or to test bidding against auction listings, you should create at least three test users: one test seller and two test buyers.

Depending on the needs of your application, you may also need to perform these steps:

- To make API calls in the Sandbox on behalf of test users, create Sandbox authentication tokens (*auth tokens*) for each test user as needed. See “[Generating Auth Tokens for Test Users](#)” on page 65. For example:
 - If your application will list items in the Sandbox, then the test user who will be “selling” those items needs a Sandbox auth token.
 - If you will only use the Sandbox UI to test buying your test seller’s items, then your test users who are buyers don’t necessarily need Sandbox auth tokens.
 - If you are creating a buying application (one that uses [PlaceOffer](#)), then the test buyers who use your application need Sandbox auth tokens.
- If your test users need to use PayPal in the Sandbox, see “[Creating a PayPal Sandbox account](#)” on page 199.

Tips on Working with Test Users

For test users to perform actions associated with a test run of an application, eBay allots a pre-determined amount of credit (play money) to each test user’s account. The accounts are automatically refreshed to \$500,000 each weekend. This means you can safely retry listing items individually and in bulk without having to pay real fees to eBay, and without any obligation to deliver real items to real buyers.

While an application might only supply functionality that serves sellers (listing items only, no bidding or purchasing), you should test the effects of buyer actions to ensure that the application handles all of the interactions the application may encounter when in production.

In general, even if your application just lists items, you should test it for typical scenarios such as users bidding on items, users winning bids, the calculation of shipping costs, and the non-paying bidder.

Likewise, if you are creating an application that caters only to buyers, you should have at least one test user designated as a seller. After all, a buyer cannot bid on an item unless a seller has listed it.

Generating Auth Tokens for Test Users

Your application needs authentication tokens to make authenticated and authorized eBay API calls on behalf of your users.

This section describes the easiest way to create an auth token for a test user.

Note: If you’re looking for information about creating and using authentication tokens for *real* users (in the Production environment), see “[Getting Tokens](#)” on page 117.

To generate an Auth Token for a registered test user, you can follow the procedure described below, or use the *Getting Tokens API Flow Tutorial*, which includes links to examples.

Getting Tokens API Flow Tutorial

<http://developer.ebay.com/DevZone/XML/docs/HowTo/Tokens/GettingTokens.html>

Here is brief summary of the procedure:

- 1 Log in to the eBay Developers Program site using your normal User ID and password:

<https://developer.ebay.com/DevZone/account/>

- 2 Click the **Get a User Token** link under the Tools section at the bottom-left of the page.

- 3 Select **Sandbox** as the environment, then select the **Key Set 1** to use for this user.

Before you can select the Sandbox key set, you must have generated a Sandbox key set for your normal User ID. If you have done this, you should be able to select **Key Set 1**, and the **DevID**, **AppID**, and **CertID** key fields should populate.

Note that test users do not have their own set of keys. Test user keys are associated with the key set that belongs to the User ID that created the test user. This is important because some calls (such as [ConfirmIdentity](#)) require that you use a key set in the request.

- 4 Click the **Continue to generate token** button.

The *Sign in to Link Your eBay Account* page displays.

- 5 Enter the User ID and password for the test user for whom you want to generate an Auth Token.

The test user's User ID usually begins with “**TESTUSER_**”.

- 6 The *Token Generation – Final Step* page displays, providing the test user's Auth Token and token expiration date.

Save the Auth Token and expiration date to a place where you can access them whenever you want to make calls on behalf of the respective test user.

After generating individual authentication tokens for each user, place the respective test user's Auth Token in the **eBayAuthToken** field of your API request. Make sure to use the proper Sandbox endpoint, depending on the format of your request. See the *Getting Tokens API flow tutorial* listed above for examples.

Testing Your Application's API Access

Some API calls have access rules. The rules define how many times an application can use certain API calls each day or hour. Your application also has an aggregate call limit, although not all calls are counted toward the limit.

Your application can detect and report the current API access rules and its current usage with [GetApiAccessRules](#), which shows which API calls have limits, what the limits are, and whether they count toward the aggregate. Each [ApiAccessRule](#) defines a rule, and you can parse for various values.

In your application, you can check for the hourly or daily usage of a specific call, say [AddItem](#). Check your application's API access when you test it in the Sandbox, and also while your application runs on the Production server. Call rate limits are set at the sole discretion of eBay. When your application goes through the Compatible Application Check, your application will be rate-limited on an application basis or a per-user basis, depending on your customer base and distribution format. Application-based call limits mean that your application will have a maximum API call cap on a daily and monthly basis. The user-based call limits mean that each user of your application gets their

own pool of API calls. One person's usage does not affect another's, so as many people can use your program as you'd like.

To learn more, refer to the Compatible Application Check page on the eBay Developers Program site:

<http://developer.ebay.com/support/certification>

If you decide to go through the Compatible Application Check, ensure that your application complies with eBay policies:

<http://developer.ebay.com/join/policies/>

Creating Stores in the Sandbox

If you want to test your application's ability to list items in an eBay Store, you need to set up a storefront in the Sandbox. See "[Creating a Store](#)" on page 535.

For conceptual information on eBay Stores, see "[eBay Stores](#)" on page 154.

Checklist for Going Live

Your application may need to go through the Compatible Application Check. For more information, please see the sections below and see the following location:

<http://developer.ebay.com/support/certification>

If you decide to go through the Compatible Application Check, ensure that your application complies with eBay policies:

<http://developer.ebay.com/join/policies/>

Compatible Application Check formerly was called Standard Application Certification.

Additionally, new applications must support the latest schema version that is available in the Production environment at the time of the Compatible Application Check.

The following topics provide more information about these prerequisites:

["Compatible Application Check"](#) on page 67
["Supporting Compatibility Levels"](#) on page 71

Compatible Application Check

Compatible Application Check formerly was called Standard Application Certification.

This section covers the following topics:

- “[Requirements for Compatible Application Check](#)” on page 68
- “[Getting a Compatible Application Check](#)” on page 70
- “[About the Compatible Application Check Timeline](#)” on page 71

Requirements for Compatible Application Check

As part of the Compatible Application Check process (see “[Getting a Compatible Application Check](#)” on page 70), eBay critically reviews all new applications for compliance with eBay’s Compatible Application Check Requirements at no cost.

The requirements are listed below. They identify performance criteria for each application. These criteria affect key operating functions: Error handling, simultaneous requests, and polling frequencies.

All Developers Program licensees are required to be in compliance with the Compatible Application Check Requirements at all times. Occasionally, changes to the API require changes to the requirements. The eBay Developers Program makes all reasonable attempts to notify licensees when the requirements change. However, we strongly recommend that you review the Compatible Application Check Requirements below regularly.

Requirements for Specific Calls

Certain calls have special requirements to ensure that you use them most efficiently and effectively. Please see “[Compatible Application Check for Specific Calls](#)” on page 963 for a summary of the Compatible Application Check Requirements that apply to particular calls.

Requirements for Error Handling

Errors that are returned from calls include useful details and instructions for both end-users and applications. The structure and granularity of the error messages are intended to help your application handle errors gracefully.

When an error occurs, please use the following guidelines to determine what actions to take to correct the error and when to retry the request.

Infrastructure Errors: These indicate that an infrastructure error has occurred, such as a problem on eBay’s side with a database or server going down.

- **Symptoms:** No response, unable to connect to eBay, a database error is returned, or a structured exception is returned.
- **Retry:** two times
- **Actions:**
 - Test connectivity.
 - Check the Announcement Boards (see “[System Announcements](#)” on page 37).
 - Escalate the problem to eBay by contacting online technical support (see “[Developer Technical Support](#)” on page 38).

Application-Level Errors: An application-level error occurs due to problems with business-level data. In these cases, we return a list of errors in the response message payload. See “[Application-Level Errors](#)” on page 108 for details about this type of error.

- **Symptoms:** An API call request returns an error result set.
- **Retry:** Do not retry until you have confirmed that the input arguments and request are valid.
- **Actions:**
 - Review the call input to confirm that the input fields and overall format of the request are valid.
 - Report bugs by using online technical support (see “[Developer Technical Support](#)” on page 38).
 - Check the Announcement Boards see “[System Announcements](#)” on page 37).
 - Escalate the problem to eBay by contacting online technical support (see “[Developer Technical Support](#)” on page 38).

See “[Error Handling](#)” on page 107 for additional information about handling errors.

Requirements for Sending Simultaneous Requests

- Server-side applications may use a maximum of 18 simultaneous threads per application. (Technically, this means 18 simultaneous threads per *AppID*, which is a unique identifier for an application. See “[Sandbox and Production Keys](#)” on page 71 for information about obtaining an *AppID*.)
- Client-side applications may use a maximum of three simultaneous threads per user.
- Any multi-threaded application must have the ability to quickly increase and decrease the thread count per call.

If you have questions about the maximum number of threads your application is permitted, please contact Developer Support.

Multiple Items Returned per Single API Call

Some calls return multiple items in response from a single call. eBay reserves the right to change the maximum number of items returned per page (see “[Controlling the Amount of Data Returned](#)” on page 100 for information about pagination).

Applications must be designed to recover gracefully when the number of items returned per page changes.

Polling Frequencies

eBay reserves the right to restrict an application’s polling frequencies for any calls. This type of restriction typically occurs when eBay determines that an application is causing a negative operational impact to the eBay site.

Versions

New applications must comply with and use the latest schema version that is available in the production environment at the time of the Compatible Application Check. See our [eBay Schema Versioning Strategy](#) for details.

We announce important changes for each version in the release notes. Therefore, it's a good idea to be sure you review the release notes regularly. See "[Release Notes and Known Issues](#)" on page 37 for information about where to find the release notes.

UTF-8 Character Set

Applications must use the UTF-8 encoding scheme (not ISO-8859-1). All applications must validate text strings specified in the body of each request to make sure the characters are UTF-8 compliant. It is not enough to simply transmit information to eBay in UTF-8 format; you must convert the characters provided by users (e.g. in item descriptions, street addresses, names of persons) to UTF-8 if they were received in a non-UTF-8 charset (e.g. from a browser for which the encoding was not UTF-8) or stored in your database in a non-UTF-8 charset.

See the article "*Working with UTF-8: charsets, conversion, tools*" in the Developers Program Knowledge Base for additional information:

<http://developer.ebay.com/DevZone/support/KnowledgeBase.asp>

Getting a Compatible Application Check

The details of the Compatible Application Check process, and best practices, are detailed on the Compatible Application Check page of the eBay Developers Program site:

Compatible Application Check Page
<http://developer.ebay.com/support/certification>

If you decide to go through the Compatible Application Check, ensure that your application complies with eBay policies:

<http://developer.ebay.com/join/policies/>

This section summarizes the Compatible Application Check process and includes some points you should keep in mind as you put together your project plan and milestones, and as you design, develop, and test your application.

These are the main steps in the Compatible Application Check process:

- 1 Double-check that your application meets the Compatible Application Check Requirements (see "[Requirements for Compatible Application Check](#)" on page 68). Conditions change and the requirements are updated on an ongoing basis.
- 2 Navigate to the Compatible Application Check page on the eBay Developers Program site (see the link above).
- 3 Navigate to the Support page on the eBay Developers Program site (see "[Developer Technical Support](#)" on page 38) and open a new support case. Fill out the Compatible Application Check

form completely and accurately. This is the only way to request the Compatible Application Check for your application.

This process is free.

- 4 After you submit the document, Developer Technical Support will notify you of the status of your application.

If your application does not initially pass the compatibility check, Developer Technical Support will explain what you need to do to improve the application. Depending on the extent of the changes, we may need to close the case, and you may need to reapply when your application is ready.

- 5 Your application must comply with the Compatible Application Check Requirements at all times, so please be sure to review the requirements regularly to make sure your application remains in compliance.

About the Compatible Application Check Timeline

The Compatible Application Check process takes approximately 5-7 business days, provided your application meets the Compatible Application Check Requirements, and you are within the API Public License Agreement guidelines. For more information, see:

<http://developer.ebay.com/support/certification>

If you decide to go through the Compatible Application Check, ensure that your application complies with eBay policies:

<http://developer.ebay.com/join/policies/>

Supporting Compatibility Levels

When you send information to eBay, you need to specify which version of the API your application is using (see “[Release Versions](#)” on page 84).

See our [eBay Schema Versioning Strategy](#) for details.

Sandbox and Production Keys

For an application to be able to operate in the Production or the Sandbox environment, it needs to have the appropriate IDs for that environment. These IDs are referred to as *development keys* (because they unlock the door into the particular development environment). Development keys consist of a set of data that identifies the application and its developer. You use these keys when you generate an authentication token for a user.

When you join the eBay Developers Program, you are provided with key sets for your application. The keys set for the Sandbox is different from the key set for the Production environment. (Keys for the Sandbox cannot be used to make API calls in the Production environment. Conversely, Production keys cannot be used to make API calls in the Sandbox.)

Please see “[Getting a Compatible Application Check](#)” on page 70.

Your keys consist of three IDs:

Table 4-3 Development Keys

Value	Meaning
DeVID	Unique identifier for the developer's (or company's) account.
AppID	Unique identifier for the application.
CertID	Certificate that authenticates the application when making API calls. Not to be confused with user-level authentication tokens. See " Security " on page 101.

The keys are created by and maintained at eBay. You cannot choose your own keys. To retrieve your keys at any time, you can use the following location:

Account Information (requires login)
<http://developer.ebay.com/DevZone/account>

You can view and track an application's API usage with the API Usage Report . (Note that multiple AppID/CertID pairs can be issued for a single DeVID.)

Your API Usage Report (requires login)
<http://developer.ebay.com/devzone/account/dashboard>

If you ever lose your keys you can retrieve them at the following location on the Developers Program Web site.

Retrieve Development Keys (requires login)
<http://developer.ebay.com/devzone/account/>

For an example of where to use these keys in an application, see "[Executing Your First API Call](#)" on page 55.

When you execute an API call, your request needs to pass these basic security checks:

- Authenticate your application by specifying appropriate development keys with your API request.

Just as you would store and protect passwords, you should also exercise the same caution in storing and using your development keys. For example, if the strings are hard-coded in a compiled application, it is possible for an unscrupulous person to see the IDs by inspecting the executable file with a tool such as a hexadecimal file viewer. We highly recommend some form of encryption of the IDs in compiled applications to deter this type of unauthorized access.

- Authenticate the user by specifying a secret *authentication token* in each API request.

The token is equivalent to the user signing in on the eBay Web site. It also indicates that the user has authorized your application to interact with eBay on their behalf.

When you initially ask eBay to generate an authentication token for a user, you must provide your development keys *and* the user must sign in to eBay and give their consent to authorize

your application to perform certain actions on their behalf. The token value is generated based on the user's sign-in credentials and your application's credentials.

From then on, your application passes both the token and the matching development keys in each API request.

Once you have your development keys, see "[Getting Tokens](#)" on page 117 for information about generating and retrieving authentication tokens for each user.

Chapter 5

Invoking eBay Trading Web Services

To perform a particular task with the Trading API (e.g., list an item), you need to use a class, module, or function that creates your request, sends it to the Trading API, and handles the response.

Regardless of how the class is implemented on the client side, it needs to make the call using a particular name that is defined in the Trading API. The call names are fairly self-explanatory, like [AddItem](#)—which lists an item.

If you are using the eBay SDK for Java or the eBay SDK for .NET, you can use classes that wrap Trading API calls. Each wrapper class is named for the SOAP API call it represents. In the case of each wrapper class, “Call” is appended to the end of a SOAP API call name. For example, the SOAP API call [AddItem](#) (in the Trading API) is represented in the eBay SDK for Java and eBay SDK for .NET as [AddItemCall](#).

The schema defines the messages that you can use to access the eBay database. The API uses request-response style operations. That is, you create a request that contains business data and other instructions and you send it to eBay. eBay replies with a response that contains the data resulting from the instructions you sent in.

Note: In contrast, eBay Platform Notifications use the notification style of operation. This is a one-way push from eBay. See “[Working with Platform Notifications](#)” on page 623.

It is helpful to understand the basic design of the schema so that you know how components are defined and how to interpret those definitions. For example, in many cases we provide you with lists of valid values for fields. It is a good idea to understand how those values are represented and how to work with them.

Regardless of whether you are listing an item, browsing listings, or performing some other business task, some data is handled in a standard way for all requests. Some of this data is required with all requests. Other data is only applicable for some calls, but the way you use it is standardized. This chapter gives you an overview of this standardized data.

When a problem occurs and a request (or a portion of a request) cannot be executed for some reason, the response includes a list of errors instead of the normal business data. If the request is executed successfully but a minor problem was found or a change was made that you might not expect, the response includes the normal business data accompanied by a list of warnings. It is a good idea to understand the types of errors and warnings that eBay may return so that you can design your application to handle them gracefully. This chapter provides an overview of how to handle errors.

[“Overview of the API Schema” on page 75](#)

[“Routing the Request \(Gateway URLs\)” on page 87](#)

[“Standard Data for All Calls” on page 92](#)

[“Error Handling” on page 107](#)

[“The Object Model of the eBay SDK for Java” on page 115](#)

Overview of the API Schema

This section introduces the major components that are available in the API schema.

If you are planning to use the eBay SDK for Java or the eBay SDK for .NET, please note that the SOAP API schema contains data as it is passed from an application to eBay or from eBay to an application. This includes such code components as [Item](#), [User](#), [Feedback](#), [Order](#), and [Transaction](#). Data for these objects is set and extracted using methods or properties on those types. And the data is transferred between the application and eBay through the API call classes.

The schema is organized into these major components:

- API request and response messages (see “[Message Types](#)” on page 75)
- Core components (see “[Core Components](#)” on page 78)
- Base components (see “[Base Components](#)” on page 79, “[Data Validation](#)” on page 80, “[Code Lists](#)” on page 82)

To ensure interoperability between different platforms and clients, the schema follow certain design principles. See “[Interoperability Considerations](#)” on page 86.

Message Types

When you make a call, you send a request message to eBay and eBay returns a response message. The message types enable your application to execute common tasks that a seller would perform on eBay, such as listing items.

All the message types derive from these abstract base request and response types:

- `AbstractRequest` – This is the base type for requests.
- `AbstractResponse` – This is the base type for responses.

The abstract base types define fields that are standard for all calls. For example:

- All requests let you specify general instructions related to the data you’re sending in or the data you want eBay to return.
 - Detail level—Use this to control the amount of data returned. See “[Controlling the Amount of Data Returned](#)” on page 100.
 - Error language—Use this to specify the language you want errors returned in. See “[Customizing Your Application’s Error Handling Abilities](#)” on page 113.
 - Version—Use this to specify the version of the schema you are using. See “[Release Versions](#)” on page 84 for details.
 - Warning Level—Use this to control whether to retrieve warnings if you pass in unrecognized fields. See “[Customizing Your Application’s Error Handling Abilities](#)” on page 113.
- All response types also include standard information.
 - Version and build—Use this information to determine the version of the schema and the specific software build that eBay used when processing the request and generating the response. See “[Release Versions](#)” on page 84 for details.
 - Errors—For any request, one or more errors may be returned. Please see “[Error Handling](#)” on page 107 for more information about working with errors.
 - Acknowledgement—All responses return a standard acknowledgement element that indicates the success, failure, or partial failure of a call.
 - Time stamp—All responses return the official eBay time stamp in UTC/GMT (see “[Data Types](#)” on page 959 for information about time values). The time stamp indicates the time when eBay processed the request; it does not necessarily indicate the current eBay official eBay time. In particular, calls like [GetCategories](#) can return a cached response, so the time stamp may not be current.
- All requests support a [MessageID](#) element and all responses support a [CorrelationID](#) element. If you pass a message ID in a request (up to 64 characters), it should be unique across the eBay site. See “[Specifying a Message ID to Correlate the Request and Response](#)” on page 96 for more information. We will return the same value as the correlation ID in the response. This may be useful for tracking that a response is returned for every request and to match particular responses to particular requests.

The concrete requests and responses (e.g., [AddItemRequest](#) and [AddItemResponse](#) for the [AddItem](#) call) are derived from the abstract request types. The concrete types define context-specific business data that is applicable for the particular call. For example, [AddItemRequest](#) specifies that it takes an item ([Item](#)) as the message payload, and [AddItemResponse](#) specifies that it returns an item ID ([ItemID](#)) and fees ([Fees](#)).

The naming convention we use for the concrete type names is the name of the call followed by “Request” or “Response”:

[VerbNameRequest](#)
[VerbNameResponse](#)

For a summary of the concrete message types that are available, see “[Overview of Calls](#)” on page 770.

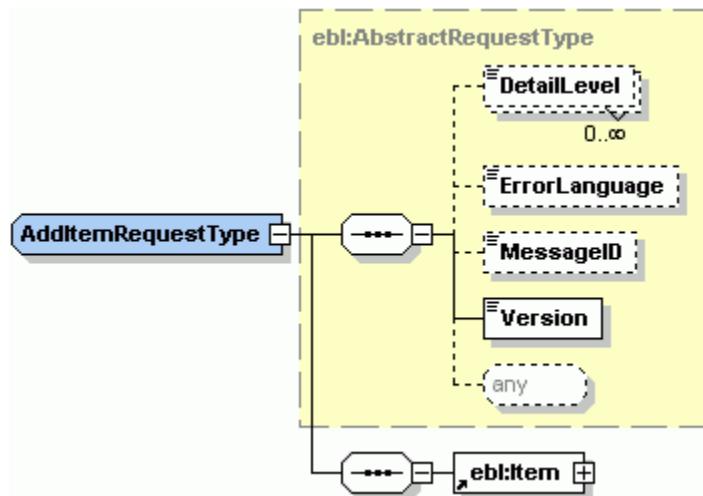
[Figure 5-1](#) on page 77 and [Figure 5-2](#) on page 78 illustrate the request and response content models.

Note: In the eBaySvc.wsdl file, which is used in the SOAP API, the message elements define the request and response messages and the data types to use in the body of those messages. The message names are the same as the data type element names (e.g., AddItemRequest).

In the portType clause, you’ll see the name of the service definition interface (the interface you’ll use to access operations) will be derived from eBayAPIInterface. This interface specifies each logical operation (e.g., AddItem) and indicates that they are bidirectional (request-response).

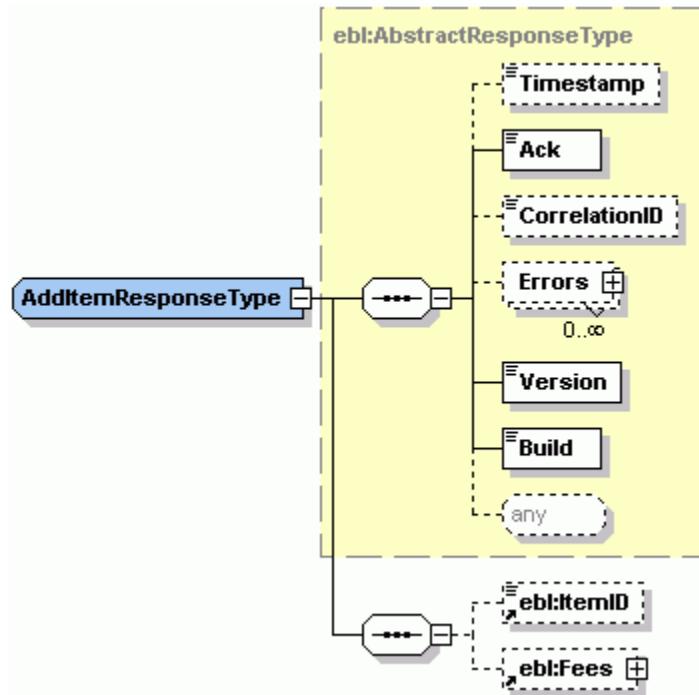
[Figure 5-1](#) shows the content model for a request (in this case, an **AddItem** request).

Figure 5-1 A Request Type



[Figure 5-2](#) shows the content model for a response (in this case, an **AddItem** response).

Figure 5-2 A Response Type



The wrapper classes in the eBay SDK for Java and in the eBay SDK for .NET use the SOAP API schema as the basis for conveying data to and from eBay.

Using an object in an SDK is slightly different from using the SOAP API directly. If you use an SDK, you create a concrete request object and populate it with data; data is entered into the properties of an SDK call class or passed as the input for an execution method. For the data returned to your application from eBay, concrete response objects are created. You may or may not need to even operate on the response object. Key data for an API call is returned by the execution method of the call class. For instance, the execution method for the `GetItemCall` class, `getItem()`, returns the data for the item in an `Item` object. The application only needs to inspect that object, rather than inspecting the response object, to get the data.

Core Components

Certain constructs, such as monetary amounts and units of measure, always require the same combination of elements in order to be semantically meaningful. For example, a monetary amount is typically modeled as a currency and a value. Various base components use amounts represent notions like listing fee amounts and account balance amounts. We define the types such as amounts as *core components* in the schema. [Example 5-1](#) is a snippet from the `FeeType` schema and shows that

the **FeeType** component consists of a name (a string describing what the fee is for) and a fee (e.g., USD 1.00). The fee is represented as an amount type,

Example 5-1 Definition of a Type that Uses a Core Component (XML Schema)

```
<xs:complexType name="FeeType">
  <xs:sequence>
    <xs:element name="Name" type="xs:string" minOccurs="0" />
    <xs:element name="Fee" type="cc:AmountType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

Example 5-2 shows what this core component data would look like in an actual XML document (e.g., in the call response).

Example 5-2 Data for a Type that Uses a Core Component (XML)

```
<Fees>
  <Fee>
    <Name>AuctionLengthFee</Name>
    <Fee currencyID="USD">0.0</Fee>
  </Fee>
  <Fee>
    <Name>BoldFee</Name>
    <Fee currencyID="USD">0.0</Fee>
  </Fee>
  ...
  ... more Fee nodes ...
  <Fee>
    <Name>SubtitleFee</Name>
    <Fee currencyID="USD">0.0</Fee>
  </Fee>
</Fees>
```

See “[Listing Items](#)” on page 174 for information about working with these types when you list and retrieve items. See “[eBay Types](#)” on page 959 for information about other supported data types.

Base Components

eBay’s base components are simple and complex types that model eBay business-level objects (e.g., items) as well as eBay API-specific notions (e.g., detail levels). The abstract request and response types are also base types (see “[Message Types](#)” on page 75).

When you design your application's business logic, these are some of the most important eBay components to understand:

- [Item](#) (see “[Items and Listings](#)” on page 136)
- [Category](#) (see “[Categories \(Introduction\)](#)” on page 138)
- [User](#) (see “[Users](#)” on page 145)
- [Order](#)
- [Transaction](#) (see “[Order Line Items](#)” on page 145)
- [Feedback](#) (see “[Feedback](#)” on page 145)

Each of the above components contains a number of subcomponents. For example, [Item](#) contains a number of objects that describe the details of the listing itself, such as [Title](#) and [Description](#), as well as other business-level information.

Many subcomponents are defined as simple types in the schema. For example, [Item.Title](#) is a simple type that just takes a string value. Some components are complex types that group sets of data into logical subcomponents, according to their usage. For example, an [Item](#) can contain a [ShippingDetails](#) object, which contains a number of other components.

The schema (XSD or WSDL) provides a low-level view of how each component is modeled (see “[Where to Find the eBay Schema Files](#)” on page 51), or refer to the [eBay Trading API Call Reference](#).

Data Validation

When you work with the API, it is important to understand the rules eBay uses for validating input fields and when fields are returned in a given call's response.

Predefined Values

Some types have a predefined list (enumeration) of valid input and output values. To help you reduce errors due to invalid input data, the schema provides you with the valid values. These values are defined in *code lists*. For example, [ListingDuration](#) is a code list that specifies all the valid listing durations on eBay. See “[Code Lists](#)” on page 82 for more information about code lists. If a type uses a code list, only the values in that list can be passed in, and they must be spelled in the same way as they are spelled in that list.

Note: A few components do not have corresponding code lists. Instead, the valid values are described in the documentation of the element itself. For example, [StoreCategoryID](#) has no code list. Its list of possible values is described in the element's type documentation instead.

Applicability for Each Call

Some components (e.g., [TimeLeft](#)) are only applicable when an [Item](#) is returned with a [GetItem](#) response; but they are invalid or ignored when an [Item](#) is passed in an [AddItem](#) request. The applicability of each field in the context of each call is described later in this guide (e.g., “[Listing Items](#)” on page 174), as well as in the [eBay Trading API Call Reference](#). Also see “[Controlling the Amount of Data Returned](#)” on page 100 for information about the ways in which you can manipulate the set of components that are returned. In many cases, a field or value that is not applicable for a call is ignored if it is passed in. However, if passing a particular value or field would cause the intent of

the request to become ambiguous or inconsistent, the call will return an error. For example, if you pass both flat-rate shipping fields and calculated-rate shipping fields when you list an item, the call will fail.

Required and Optional Fields

As a general convention, all elements in the schema are defined with a multiplicity of `minOccurs="0"`. This does not necessarily mean the field is optional in a call you are using.

Note: If an element is defined with a multiplicity of `minOccurs="1"` (or no `minOccurs` value, which means it defaults to 1), then the field is required in all cases. However, most elements in the eBay schema are defined with a multiplicity of `minOccurs="0"`.

The schema uses this convention for few reasons:

- The same types can be shared across multiple calls, but some elements defined on those types might not be shared. For example, `Item.Country` is required as input to `AddItem`, but it isn't applicable in the response of `GetSellerEvents`. As both of these calls share the same basic `Item` object, eBay needs to set the `Item.Country` element's multiplicity to `minOccurs="0"` in the schema so that `Item` can be used in both cases.
- The same field can be required in one use case and optional in another, even within the same call. For example, in `AddItem`, the `Item.Title` field is required as input in most cases, but it's optional if you use a feature that "pre-fills" the title for you. Therefore, eBay needs to define the `Item.Title` element's multiplicity as `minOccurs="0"` in the schema to account for this "pre-filling" use case.
- Future eBay enhancements can change a required field to optional. For example, `Item.Location` used to be required in `AddItem`. But when we added a new `Item.PostalCode` field (to support a new search feature), the `Location` field became optional. Therefore, to "future-proof" the schema against such logical changes, eBay usually defines fields as `minOccurs="0"` in the schema, even when the field is currently required.

To determine whether or not a field is required in a particular context, please refer to the [eBay Trading API Call Reference](#).

If any required fields are missing when you use a call, the request will fail and return errors. See "[Error Handling](#)" on page 107.

Repeating (Unbounded) Fields

If the multiplicity of an element is defined as `maxOccurs="unbounded"` or a value greater than 1 in the schema, it means you can specify the element multiple times in the request (like you are specifying an array). If you do so, the fields must be contiguous. That is, you cannot insert other fields between the repeating fields. Otherwise, some of the data will be dropped in an unpredictable manner.

For example, **AddItem** lets you specify multiple payment methods in the request. Here is the definition of the **PaymentMethod** element and another element called **PayPalEmailAddress** in the **Item** schema (**ItemType**):

```
<xs:element name="PaymentMethods" type="ns:BuyerPaymentMethodCodeType"
minOccurs="0" maxOccurs="unbounded" />
<xs:element name="PayPalEmailAddress" type="xs:string" minOccurs="0" />
```

When you send the request to eBay, the **PayPalEmailAddress** element must be inserted after all the **PaymentMethod** elements, not between them:

```
...
<PaymentMethods>CashOnPickup</PaymentMethods>
<PaymentMethods>PayPal</PaymentMethods>
<PayPalEmailAddress>mypaypalaccountemail@ebay.com</PayPalEmailAddress>
... other elements here ...
```

Length Restrictions

All string input fields have length restrictions. Some string input fields (e.g., **Item.SubTitle**) do not have a known set of possible values. For these arguments, the documentation specifies the maximum permitted length in characters. Other string fields (e.g., **Country**) do have a known set of possible values. In these cases, the maximum length is not specified in the documentation. Fields of other types (e.g., integers) may also have length restrictions. These restrictions are also specified in the documentation, where applicable.

Data Types

The schema enforces the usage of correct data types. For example, if an element is defined as a double, you need to pass in a numeric value. See “[Data Types](#)” on page 959.

Code Lists

Some base components have predefined lists of possible input and output values (e.g., listing durations). Each code list is an enumeration of values. For example, the **ListingDuration** code list contains all the possible listing durations (e.g., **Days_3**) on eBay. If a field uses a code list to define its possible values, you can only pass those values in to the call. If you send in an undefined value (including a misspelled value), an error is returned.

The naming convention we use for the code list names is the name of the component followed by “**CodeType**”:

ComponentNameCodeType

Example 5-3 is a snippet from the `ListingDurationCodeType` schema and shows that the `ListingDuration` codes have values like "Days_1" for one-day listings.

Example 5-3 Portions of a Code List (XSD)

```
<xs:simpleType name="ListingDurationCodeType">
  <xs:annotation>
    <xs:documentation>
      ListingDurationCodeType - Type declaration to be used by other schema.
      Each code specifies a number of days that a listing can be active
      (i.e., available for bidding/buying). The validity of a code depends on the listing type.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:enumeration value="Days_1">
      <xs:annotation>
        <xs:documentation>
          (in/out) 1 Day - A seller must have a positive feedback rating of 10
          or more or must be ID Verified to use the 1-day listing duration on
          the US site. Applicable for Chinese auctions
          (including Real Estate) and Personal Offer (Second Chance Offer)
        </xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <!-- ... More codes here ... -->
    <xs:enumeration value="CustomCode">
      <xs:annotation>
        <xs:documentation>
          (out) Reserved for internal or future use
        </xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

For a SOAP API example showing portions of the same code list in a generated stub, see the `ListingDurationCodeType.java` file in the eBay SDK for Java:

eBay SDK for Java
http://developer.ebay.com/community/featured_projects/?name=eBay+SDK+for+Java

Example 5-4, which also applies to the SOAP API, shows portions of the same code list in a generated stub class in C#. Each code list is an enumeration defining a set of named constants.

Example 5-4 Portions of a Code List Stub (C#)

```
public enum ListingDurationCodeType {
  Days_1,
  // ... more enumeration elements here ...
  CustomCode,
}
```

Some codes are valid in requests and responses, some are only valid in requests, and others are only returned in responses. In the schema, the usage is specified in the code list annotations (see [Example 5-3](#) on page 83):

- **(in, out)** means the value can be specified as input and can be returned in responses.
- **(in)** means the value can be specified as input, but it is not returned in responses. Usually, this occurs when a computed value is returned in responses instead. For example, you specify a listing duration for an listing when you use [AddItem](#). But, the [GetItem](#) response does not return the listing duration. Instead, it returns the start time and the end time of the listing.
- **(out)** means the value cannot be specified as input. It is only returned in responses.

Very important: All code lists support a default value (`CustomCode`). If your application uses a supported but outdated version of the schema, it is possible that you will retrieve data where a user has selected an option (on the eBay site or via another application) that your application does not recognize. When your application sends a request, eBay checks the version you pass in. If a value in the response doesn't exist in that version, we map the undefined value to `CustomCode`. If you detect this value in a response message, you should update your version of the schema to get the latest code lists. See "[Where to Find the eBay Schema Files](#)" on page 51.

Note: This `CustomCode` convention is necessary because SOAP clients cannot easily tolerate new or unexpected enumerated values. If a SOAP client only supports values A, B, C, and we start sending D to that client, it will throw an illegal state exception. In order to have the type safety of enumerations but allow clients that still use older schema versions to continue to work, we have to coerce the new "D" value to a default value that the client understands. This is why all enums have `CustomCode`.

When you work with code lists, you will need to manage the eBay business logic that affects your application. Code lists define all possible input and output values generically. That is, they do not describe business rules and dependencies. For example:

- Certain arguments may become valid or invalid according to other values specified in the request. For example, the [ListingDuration](#) code list does not programmatically specify which durations are valid for Chinese Auction listings. Instead, this kind of information is described in the code list annotations.
- Certain values may become required when other values are specified in the request. For example, if you specify a shipping option, you must also specify a shipping region. This kind of information is available in this documentation (see "[Listing Items](#)" on page 174).

Release Versions

The release version is the version number of the eBay Trading API that you are programming against (e.g., 813). See "[Where to Find the eBay Schema Files](#)" on page 51 to download the latest schema.

Every two weeks, eBay updates features and adds new ones. Sometimes the update affects everybody no matter what version they are using, and sometimes an update only affects a particular API version. So it's important to keep track of what version of the API you're using.

If you are just starting out, start with the latest version, and always make sure your application is using at least the lowest supported version.

Each time you execute a request, you need to specify the version of the API you are using. Also, although you are not required to upgrade your application each time a new version of the API is released, it is a good idea to at least be aware of changes made in each release.

Some changes may affect your application's business logic or ability to execute calls successfully. The version number you specify in the gateway URL and request body indicates the schema version that your application is using. If you are an SDK user and are considering updating your SDK to a more recent version, see the Readme file for your SDK.

This section covers the following topics:

["The eBay Developers Program Release Cycle"](#) on page 85

["When to Update the Schema"](#) on page 86

The eBay Developers Program Release Cycle

When you send a request to eBay, you need to specify the version of the schema that you are using (see ["Routing the Request \(Gateway URLs\)"](#) on page 87 and ["Specifying the Schema Version"](#) on page 95).

Note: You pass the version in the **Version** field, and the same number is returned in the response in a **Version** field. Each response also contains a **Build** field. This refers to the specific software build that eBay used when processing the request and generating the response. Developer Support may request the build information when helping you resolve technical issues.

If you are using the eBay SDK for Java, you can determine the WSDL version by creating and setting up an **ApiClient** object and inspecting the **WSDLVersion** property (using the **ApiClient.getWSDLVersion()** method). Compare that version with the version on the eBay Developers Program site if you are considering whether to download a new WSDL version.

If you are using the eBay SDK for .NET, you can determine the WSDL version by creating and setting up an **ApiClient** object and inspecting the **Version** property. Compare that version with the version on the eBay Developers Program site if you are considering whether to download a new WSDL version.

When a new version of the API becomes available, we announce it in the newsletter (see ["eBay Developers Program News"](#) on page 36). You can also check the eBay Developers Program site periodically to find out if any new versions are available. The latest version of the API is identified by the version number specified in the latest eBay schema file.

New features and functional changes are initially available on US sites (e.g., ebay.com and stores.ebay.com). We release functionality applicable to international sites (e.g., ebay.co.uk) about one week after the US site release. This means that when we announce new functionality in the Newsletter or Release Notes, that functionality is only available on the US site. It will be available to international sites about one week later. For a list of supported international sites, see the site code list (**SiteCodeType**) and ["Field Differences for eBay Sites"](#) on page 590.

Note: For communication purposes in the newsletter and release notes, we add an "i" suffix to version numbers to clarify which versions are in effect internationally (e.g., "Version 695i"). The "i" is not used in the schema or the URL to the schema, and you should not pass the "i" when submitting requests to international sites.

When to Update the Schema

Older versions of the schema are periodically deprecated, which means they are no longer supported. See the [eBay Trading API Schema Versioning Strategy](#) for details about how deprecation works for the Trading API and a list of supported versions.

If your application uses an outdated version of the eBay schema, eBay will still process the request as long as that version is at or above the lowest supported version.

It is a good idea to test code list values in the response payload to make sure they don't contain the value "CustomCode" (see ["Code Lists"](#) on page 82). If you see this value, it's a signal that a code list you're using is out of date with the version on the server.

You should plan to update your version of the schema if you want to support the latest data that eBay has specified. See ["Where to Find the eBay Schema Files"](#) on page 51 for information about updating your local version of the eBay Trading API schema.

Interoperability Considerations

To reduce interoperability issues (due to differences in SOAP client implementations for different languages and operating environments), we have made certain design decisions that may affect how you design your application to use the base and core component types:

Polymorphism: We limit extension to the request type schema and response type schema. For example, `AddItemRequest` extends `AbstractRequest`. In all other cases, we avoid polymorphism. For example, when specifying shipping details (`ShippingDetails`), you need to choose either a flat rate or a calculated rate. Instead of extending a common rate type, we define separate `FlatShippingRate` and `CalculatedShippingRate` containers. You can specify either one in the shipping details, but not both. Your application will need to handle such business logic (the "or" logic is not exposed in the schema). For details about such dependencies, please see ["Listing Items"](#) on page 174 and related topics.

Backward Compatibility: We use a versioning system so that your applications will be backward compatible when new elements appear in the server-side schema. See ["Interoperability Considerations"](#) on page 86 and the [eBay Schema Versioning Strategy](#). We also provide hints to help developers determine when certain data is out of date on the client side. See ["Code Lists"](#) on page 82.

Cardinality/Multiplicity: For base components, the cardinality of major containers is set to `minOccurs="0"` (but `maxOccurs` can vary). This will allow us to reuse the same container objects across different use cases that might require different combinations of child components in the future.

In addition, we adhere to the following rules for representing data types in the schema:

- For numeric values, we use the int (32-bit) data type instead of long (64-bit), and float instead of decimal for percentage values.
- We derive the core monetary amount type from double.
- We return time values in GMT/UTC, using the ISO format. However, some development environments (e.g., the Java 2 Platform and the Microsoft .NET Framework) convert time values returned in SOAP messages to the time zone specified in your environment's locale.

See “[eBay Types](#)” on page 959 for a list of data types and information about working with time values.

Routing the Request (Gateway URLs)

All API requests that you execute must be sent to eBay’s Sandbox or Production API Gateway. You specify the gateway URI in the request. If you are using the SOAP API, you append a query string that tells eBay how to route your request.

The Gateway URIs for Sandbox data and API calls are:

XML API Sandbox Gateway URI
<https://api.sandbox.ebay.com/ws/api.dll>

SOAP API Sandbox Gateway URI
<https://api.sandbox.ebay.com/wsapi>

The Gateway URIs for live, production data and API calls are:

XML API Production Gateway URI
<https://api.ebay.com/ws/api.dll>

SOAP API Production Gateway URI
<https://api.ebay.com/wsapi>

We require you to use HTTPS/SSL when you execute calls in the Production environment. This ensures that data can be transferred securely between your application and the eBay platform. Although Sandbox data is simulated, you also need to use HTTPS/SSL when you execute calls in the Sandbox.

Note that although some of your developer keys are optional for certain calls, the full set of developer keys is required for GetTokenStatus and RevokeToken API calls. The tables in this section will give you more information about when specific information is required for a call.

See “[Standard Data for All Calls](#)” on page 92 for the data required in all Trading API requests. See “[The Sandbox and Production Environments](#)” on page 149 for general information about working in the Sandbox and Production environments.

Note: In the WSDL for the SOAP API, the service locator specifies eBay’s Production SOAP gateway URI (the physical location of the service endpoint). In the service clause of the WSDL, you can see that the name of the service locator will be derived from `eBayAPIInterfaceService`. However, as the same WSDL is used both in the Sandbox and in Production, you should pass in the service endpoint yourself rather than obtaining it from the service locator.

HTTP Headers (XML API Only)

To properly route an XML API request to the appropriate destination within eBay, pass the following information in the HTTP headers. See “[Executing Your First API Call](#)” on page 55 for an example of

how to do this. (If you're using the SOAP API, .NET SDK, or Java SDK, see “[SOAP URL Parameters \(SOAP API and SDKs Only\)](#)” on page 89.)

Table 5-1 HTTP Headers for XML API Calls

Name	Required?	Description
X-EBAY-API-COMPATIBILITY-LEVEL	Always	The eBay release version that your application supports. See the eBay Schema Versioning Strategy for information about how the version affects the way eBay processes your request.
X-EBAY-API-DEV-NAME	Conditionally	<p>Your Developer ID (DevID), as registered with the eBay Developers Program.</p> <p>The developer ID is unique to each licensed developer (or company).</p> <p>This value is only required for calls that set up and retrieve a user's authentication token (these calls are: GetSessionID, FetchToken, GetTokenStatus, and RevokeToken). In all other calls, this value is ignored..</p> <p>If you lose your keys you can retrieve them using the View Keys link on your My Account page. Here is the direct link to the Keys page (requires login):</p> <p>http://developer.ebay.com/DevZone/account/keys.asp</p>
X-EBAY-API-APP-NAME	Conditionally	<p>Your application ID (AppID), as registered with the eBay Developers Program.</p> <p>This value is only required for calls that set up and retrieve a user's authentication token (e.g., FetchToken). In all other calls, this value is ignored. Do not specify this value in AddItem and other calls that list items.</p> <p>The application ID is unique to each application created by the developer.</p> <p>The application ID and certificate ID are issued in pairs. Multiple application/certificate ID pairs can be issued for a single developer ID.</p>
X-EBAY-API-CERT-NAME	Conditionally	<p>Your certificate ID (CertID), as registered with the eBay Developers Program.</p> <p>This value is only required for calls that set up and retrieve a user's authentication token (e.g., FetchToken). In all other calls, this value is ignored. Do not specify this value in AddItem and other calls that list items.</p> <p>The certificate ID is unique to each application created by the developer.</p>
X-EBAY-API-CALL-NAME	Always	<p>Canonical name of the call you are using (e.g., AddItem).</p> <p>To be routed correctly, the value must match the request name. For example, if you are using AddItemRequest, specify “AddItem” (without “Request”) in this call name header.</p>

Table 5-1 HTTP Headers for XML API Calls

Name	Required?	Description
X-EBAY-API-SITEID	Always	eBay site to which you want to send the request. This is usually the eBay site an item is listed on or that a user is registered on, depending on the purpose of the call. See “ Specifying the Target Site ” on page 93 to understand how the site ID may affect validation of the call and how it may affect the data that is returned.
Content-Type	Recommended	For calls like AddItem, the site that you pass in the body of the request must be consistent with this header. (In AddItem, you specify the 2-letter site code. In this header, you specify the numeric site ID. See SiteCodeType for a list of valid values.) Note: If you use a library that sets the ContentType header by default, make sure it uses “text/xml”.
Content-Length	Recommended	The length of the XML request string you are sending. Used by eBay to determine how much data to read from the stream.

SOAP URL Parameters (SOAP API and SDKs Only)

To properly route a SOAP API request to the appropriate destination within eBay, pass the following information in the URL query string.

Table 5-2 URL Query String Parameters (SOAP API)

Key	Value
appid	A set of Sandbox Keys (AppId, DevId, CertId) is issued to you by Developer Relations when you first join the Developers Program. The application ID (AppId) is unique to each application created by the developer. The AppId is only applicable to calls that set up and retrieve a user’s authentication token (e.g., FetchToken). In all other calls, this value is ignored. Do not specify this value for AddItem and other calls that list items. In order to retrieve a user’s authentication token, you also will need to pass the AppId value in the SOAP envelope (along with your DevId and CertId). As mentioned above, the full set of developer keys is required for GetTokenStatus and RevokeToken API calls. See “ Security ” on page 101 for more information.
version	This is the version of the payload schema you are using. The version should match the version you pass in the message body. See “ Release Versions ” on page 84 for details.
callname	This is the canonical name of the request (e.g., AddItem). The routing service cannot look within the message body to determine the request type. Therefore, although the message itself will identify the request you’re making, this information is also required in the URL. See the eBaySvc.wsdl file or the “ Overview of Calls ” on page 770 for a list of available calls.
siteid	This is the site that item of interest is (or will be) listed on or (for requests that get/set user information) that the requesting or target user is registered on. See “ Specifying the Target Site ” on page 93 for more information about sending requests to international sites and special sites. See “ Field Differences for eBay Sites ” on page 590 for current site IDs.
Routing	Specify Routing=new to direct the call to a single namespace server.

Header recommendations: set [Content-Type](#) to “application/soap+xml”, and set [SOAPAction](#) to “”.

Theoretically, the URL resulting from the SOAP request would look something like this (if hard-coded in your application):

```
https://api.ebay.com/  
wsapi?callname=AddItem&siteid=0&appid=myappid&version=511&Routing=new
```

Note: If you use a third-party tool for monitoring or logging message data, the tool may require you to pass the URL into your application in a different manner. Please refer to the tool's documentation for instructions.

If you are using the eBay SDK for Java, then to properly route the request to the appropriate destination within eBay, specify the URL for the target eBay API server (Production or Sandbox) in the [ApiServerURL](#) property of the [ApiClient](#) object used to make the API calls. The eBay SDK for Java adds to this URL the query parameters for the appropriate API call (based on the API call class used), the target eBay site (from the [ApiClient.Site](#) property), the application ID, and the SOAP schema version.

If you are using eBay SDK for .NET, then to properly route the request to the appropriate destination within eBay, specify the URL for the target eBay API server (Production or Sandbox) in the [SoapApiServerURL](#) property of the [ApiClient](#) object used to make the API calls. The eBay SDK for .NET adds to this URL the query parameters for the appropriate API call (based on the API call class used), the target eBay site (from the [ApiClient.Site](#) property), the application ID, and the SOAP schema version.

Here is an example of the proto-URL that is specified if you are using an SDK:

```
https://api.ebay.com/wsapi
```

Below is an example of the URL that is actually used by the SDK (as described above). The [AddItemCall](#) class is being used in this case. An API call is sent to the US eBay site by an application with the app ID “myappid” using a WSDL file with a version of 511.

```
https://api.ebay.com/wsapi?callname=AddItem&siteid=0&appid=myappid&version=511
```

[Example 5-5](#) applies to the SOAP API and shows one way to specify the routing information. The query arguments are hard-coded here for illustration purposes only. This example is in Java, but the logic would be similar for any programming language.

Example 5-5 Specifying Routing Information in the Request URL

```
// Define the endpoint (e.g., the Sandbox Gateway URI)
String endpoint = "https://api.sandbox.ebay.com/wsapi";

// Define the query string parameters.
String queryString = "?callname=AddItem"
    + "&siteid=0"
    + "&appid=myappid"
    + "&version=349";
    + "&Routing=new";

String requestURL = endpoint + queryString;

EBayAPIInterfaceServiceLocator sl = new EbayAPIInterfaceServiceLocator();

EBayAPIInterface privBinding = sl.getEbayAPI(new URL(requestURL));

((EbayAPISoapBindingStub)privBinding).setTimeout(60000);
```

Note: The eBay WSDL defines the message signatures along with the service location information. See the eBaySvc.wsdl file for more information.

In the WSDL, the message elements define the request and response messages and the data types to use in the body of those messages. The message names are the same as the data type element names (e.g., AddItemRequest).

In the portType clause, you'll see the name of the service definition interface (the interface you'll use to access operations) will be derived from `EbayAPIInterface`. This interface specifies each logical operation (e.g., `AddItem`) and indicates that they are bidirectional (request-response).

In the binding clause, you'll see that the name of the proxy that implements the service definition interface will be derived from `EbayAPISoapBinding`. The binding indicates that each operation needs to be transmitted using the SOAP document-literal style.

In the service clause, you'll see that the name of the service locator in the Production environment will be derived from `EbayAPIInterfaceService`. The service locator specifies the physical location of the service endpoint (i.e., the URI of the service provider—eBay's Production SOAP API gateway). You will need to override this value when executing calls in the Sandbox. It is therefore a good idea to set the endpoint yourself in general, rather than obtaining it from the service locator.

If you are using the eBay SDK for Java, pass the URL for the target eBay API server in the `ApiClient` object used to set up the environment for a call. The API server URL is passed in the `ApiClient.ApiServerUrl` property. For example:

Example 5-6 Specifying the Request URL in eBay SDK for Java Applications

```
// Create an ApiContext object for the call
ApiClient apiContext = new ApiClient();

// Specify the eBay SOAP server URL
// without any URL parameters
apiContext.setApiServerUrl("https://api.ebay.com/wsapi");
```

If you are using the eBay SDK for .NET, pass the URL for the target eBay API server in the [ApiClient](#) object used to set up the environment for a call. The API server URL is passed in the [ApiClient.SoapApiServerUrl](#) property. For example:

Example 5-7 Specifying the Request URL in eBay SDK for .NET Applications

```
// Create an ApiContext object for the call
ApiClient apiContext = new ApiContext();

// Specify the eBay SOAP server URL
// without any URL parameters
apiContext.SoapApiServerUrl = "https://api.ebay.com/wsapi";
```

Standard Data for All Calls

Regardless of whether you are listing an item, browsing listings, or performing some other business task, some data is handled in a standard way for all requests. Some of this data is required with all requests. Other data is only applicable for some calls, but the way you use it is standardized. This section describes this standard data.

Also see “[Routing the Request \(Gateway URLs\)](#)” on page 87 for information about standard data that you need to append to the Gateway URL.

- “[Release Versions](#)” on page 84
- “[Standard Input Data](#)” on page 92
- “[Controlling the Amount of Data Returned](#)” on page 100
- “[Security](#)” on page 101

Standard Input Data

A given API call has two fundamental kinds of information you need to send: Data that is common to all API calls and data that is only used in a particular call to fulfill that call’s purpose.

In some cases, you pass this data by using input fields that are defined on [AbstractRequestType](#). In other cases, you pass this data in the URL or an HTTP header. For information on the call-specific input tags, see the chapters for the various calls in “[API Call Reference Guide](#)” on page 765. The input data common to all calls is described in the next few sections.

- “[Specifying the Target Site](#)” on page 93
- “[Specifying the Schema Version](#)” on page 95
- “[Specifying a Message ID to Correlate the Request and Response](#)” on page 96
- “[Specifying an Error Language](#)” on page 97
- “[Specifying Detail Levels and Granularity Levels](#)” on page 98
- “[Specifying Unique Identifiers \(UUID\) for Write Calls](#)” on page 99

The tags you use in an API call must match the API schema. For example, the letters in an input tag must have the case that matches the case defined for the tag in the API schema.

Specifying the Target Site

The eBay site you specify in your requests affects the business logic and validation rules that are applied to the request. For example, the eBay US site and eBay Germany site follow different rules due to differences between US and EU law, buyer behavior, and other factors.

For a list of valid site names and IDs, call [GeteBayDetails](#) or see the [SiteCodeType](#) documentation.

Site IDs

<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/types/SiteCodeType.html>

Where to Specify the Target Site (XML and SOAP)

For some requests, you need to specify the site twice:

- 1 *(All requests)* Use the numeric site ID to route the request to the correct site (see “[Routing the Request \(Gateway URLs\)](#)” on page 87). This is usually the site that the item is listed on or that the user is registered on (depending on the purpose of the call).
 - To route the request in the XML API, specify the numeric ID of the target site in the X-EBAY-API-SITEID HTTP header.
 - To route the request in the SOAP API, specify the numeric ID of the target site in the URL.
- 2 *(Some requests)* Use the site name (see [SiteCodeType](#)) in the body of the request as appropriate for the type of call you’re using.

For example, in [AddItem](#), specify the target site by using the [Item.Site](#) input field. The site you specify for the item should match the site you specify in the HTTP header or URL.

Where to Specify the Target Site (eBay SDK)

If you are using an SDK, specify the ID of the target site in the [Site](#) property of the [ApiClient](#) object that is used to make the call. The [ApiClient](#) object is reusable. An application will typically create and configure an [ApiClient](#) object once and then use it for all API calls made for the duration of an application session. This means the site ID usually only needs to be set once if you are using an SDK. Use a value in the site code list ([SiteCodeType](#)). You can change the site ID for a particular call using the [ApiClient.setSite\(\)](#) method (in the eBay SDK for Java) or using the value of the [ApiClient.Site](#) property (in the eBay SDK for .NET).

For some calls, such as [AddItem](#), you need to specify the target site when you list an item (in the [Item.Site](#) property). The site you specify for the item should match the site you specified in the [ApiClient.Site](#) property. When you set [Item.Site](#), use a value in the site code list ([SiteCodeType](#)).

Why the Target Site Matters

The site you specify in your requests affects the business logic and validation rules that are applied to the request. For example, the site can affect data like the following:

- The list of available categories (including the availability of categories that support item specifics, product details, and VAT/business items)
- International business requirements (e.g., whether a listing is subject to VAT)
- Values returned in converted prices (e.g., [Item.ListingDetails.ConvertedStartPrice](#))
- Shipping cost calculations
- Default units of measure (e.g., pounds vs. kilograms)
- Item-related time stamps that are displayed on the eBay Web site

For example, item revision times might use localized time stamps such as PDT. These time stamps are not necessarily apparent via the API, which expresses time values in GMT (see “[About GMT](#)” on page 960)

- Localization of certain standardized string values within listing details
- The language of error and warning strings that are returned (see [ErrorLanguage](#) below)
- Other data (which varies by the call you are using)

In some cases, the rules are determined by a combination of the site, the user’s registration address, and other information. (The documentation explains these dependencies on a case-by-case basis.)

The site you specify when using [AddItem](#) can also affect the visibility of the listing in some types of searches. For example, if you call [GetItem](#) for the US site, you don’t see listings that appear in categories of the UK site.

The site you specify does not necessarily act as a filter for limiting the data returned. It depends on the nature of the request:

- Item IDs are unique across all sites (i.e., an item ID on the UK site would not be re-used for a different listing on the US site). This means calls that filter data by the item ID (e.g., [GetItem](#)) retrieve the same listing no matter which site you specify. (This is important, because sometimes you’ll only know the item ID, but you might not happen to know which site the item was listed on.) In such cases, the site value you pass only affects site-specific data like [ConvertedStartPrice](#) (as described above).
- Similarly, user IDs are unique across all sites. This means that calls that retrieve data by a user ID (e.g., [GetUser](#) and [GetSellerList](#)) retrieve the data no matter which site the user is registered on

and which site you specify in the request. (Again, this is important because you may not know which site the user is actually registered on.)

- Site-specific user information (such as a subscription to Selling Manager Pro on the US site) depends on the site you specify in the request.
- As indicated earlier, category IDs are not unique across all sites, and each site has variations in the category hierarchy. Therefore, the site information has a filtering effect when retrieving category-based data and an impact on validation when you list items. Calls like **AddItem** will fail if you specify a category ID that is incorrect for the specified site. Calls that filter results by category ID (e.g., **GetCategories**) generally only retrieve data that is applicable to the site that you specify in the request.
- Other types of user information (such as registration date) does not depend on the site you specify in the request.

While the usefulness of including the **Item.Site** field in addition to the HTTP header or URL may not be externally apparent for some calls, other calls may fail or may not behave as expected if you do not pass the site data in the body or if the value you pass does not match the value in the HTTP header or URL.

Therefore, we strongly recommend that you always specify the site when you list an item. For some applications, it may be convenient to rely on reusable classes, modules, or functions to set the site ID in the HTTP header or URL and the **Item.Site** field the same time so that they always match each other.

Specifying the Schema Version

In each request, you need to pass the version of the schema you are using. How you do this depends on which format you're using:

- In the SOAP API, the version is defined on the abstract request and response types. Pass the version in the **Version** field of the request payload.
 - See “[Base Components](#)” on page 79.
 - See “[SOAP URL Parameters \(SOAP API and SDKs Only\)](#)” on page 89.
- In the XML API, the version is defined in the X-EBAY-API-COMPATIBILITY-LEVEL HTTP header (see “[HTTP Headers \(XML API Only\)](#)” on page 87).

See the [eBay Schema Versioning Strategy](#) for information about how the version affects the way eBay processes your request.

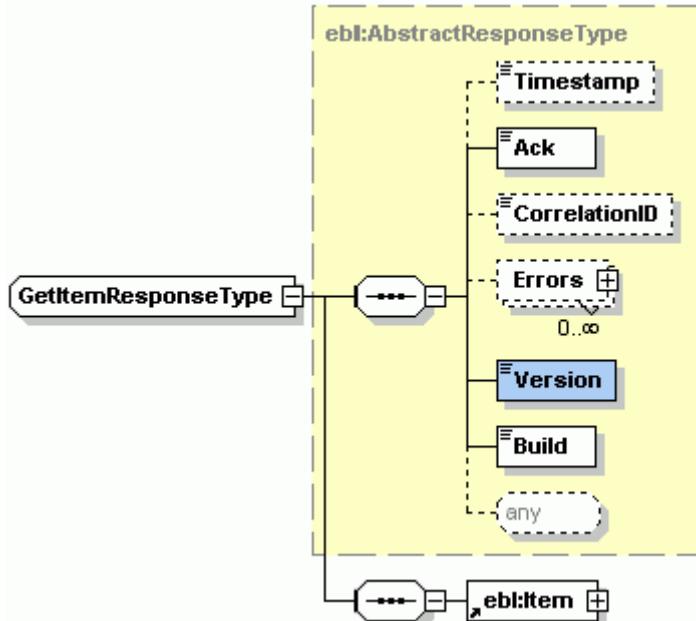
Note: To distinguish the current international version from the US version, this documentation sometimes appends the letter "i" to the release number. Please do not include the "i" when passing the version number to eBay.

The **AbstractRequestType** defines a **Version** element. However, that input element is only used by the SOAP API; it is ignored by the XML API.

eBay returns the version of the schema we used in each response's **Version** field. This lets you confirm that we used the version you expected.

Each response also contains a **Build** field. This refers to the specific software build that eBay used when processing the request and generating the response. Developer Support may request the build information when helping you resolve technical issues.

Figure 5-3 The Version Element in a SOAP Response Message



Specifying a Message ID to Correlate the Request and Response

The message ID is used to associate the response message with the particular request message that triggered it. This correlating message ID is an optional input that an application can use to enable strict tracking of request and response messages—especially important if an application has to contend with many such message pairings. Pass the message ID value in the **MessageID** property of the call's request object. The value should be unique across the eBay site and cannot exceed 64 characters.

A unique message ID should consist of a universally unique identifier (UUID) or your **InvocationID** value (see “[Specifying Unique Identifiers \(UUID\) for Write Calls](#)” on page 99) followed by a dash and an application-specific value.

UUID-ApplicationSpecificValue

The UUID or invocation ID can only contain digits from 0-9 and letters from A-F and must be 32 characters long.

Note: The **AddItem** object also accepts an **Item.UUID** value as input. When executing this call, we recommend that you use the same UUID in both the **MessageID** and the **Item.UUID**.

The value passed in **MessageID** (if any) is returned verbatim in the **CorrelationID** property of the response object. This provides the connection between a particular request and its response. (The value is not read or used in any other way by eBay, and its actual value only has meaning to the application.)

The application can use this information to definitively associate a particular response with the request that triggered it. For example, if the application serves multiple customers and has some internal identifier for each user (like the serial customer number from a table of customers) and the message ID, then the application would be able to associate responses with particular customers.

Of course, an application could use a combination of these approaches, such as passing the customer identifier within the application-specific value. That would serve to both link the request and response to each other and to associate the response with a particular customer. For example, the application might pass a value like “9CEBD9A6825644EC8D06C436D6CF494B-00451”, where the characters to the left of the hyphen are a UUID and the characters to the right are an application-specific customer ID. The maximum length is 64 characters (including the UUID and hyphen).

Many programming languages provide classes or modules that generate UUIDs.

Some calls are designed to retrieve large sets of meta-data that only change once a day or less often. To improve performance, these calls return cached responses when you request all available data (with no filters). In these cases, the correlation ID is not applicable. However, if you specify an input filter to reduce the amount data returned, you can use MessageID and CorrelationID for these meta-data calls. See the eBay Trading API Call Reference for a list of calls that can return cached data.

Specifying an Error Language

An application can specify the language in which error messages are returned. This is done through the value passed in the [ErrorLanguage](#) property of the request. Specify the standard RFC 3066 language identification tag for the language in which error messages should be returned. For example, specify `en_US` to have error messages returned in US English. Error language is an optional input which, if not specified, defaults to US English. The following table provides the language identification tag you should enter, along with the associated country and eBay site ID

Table 5-3 Error Language Codes

Language Identification Tag	Country	eBay Site ID
de_AT	Austria	16
de_CH	Switzerland	193
de_DE	Germany	77
en_AU	Australia	15
en_CA	Canada	2
en_IE	Ireland	205
en_GB	United Kingdom	3
en_US	United States	0
es_ES	Spain	186
fr_BE	Belgium (French)	23
fr_CA	Canada (French)	210
fr_FR	France	71
it_IT	Italy	101
nl_BE	Belgium (Dutch)	123
nl_NL	Netherlands	146

Table 5-3 Error Language Codes (Continued)

Language Identification Tag	Country	eBay Site ID
en_IN	India	203

Specifying Detail Levels and Granularity Levels

With many data retrieval calls that return multiple results (e.g., multiple items), you can reduce the volume of the response payload for a single API call by reducing the number of fields returned in each result.

A standard way to do this is to use a *detail level* or, for some calls, a *granularity level*. For example, a particular level might cause the response to include buyer-related data in every result, but no seller-related data. Specify a detail level using the [DetailLevel](#) property in the call request. Similarly, for calls that support granularity levels, use the [GranularityLevel](#) property of the call request.

Using a detail level (or granularity level) is just one way to reduce the size of the response payload for a single API call. For other ways, see “[Controlling the Amount of Data Returned](#)” on page 100.

The schema includes a global list of detail levels. Some calls also support an alternative [GranularityLevel](#) property. Typically, you can use either [DetailLevel](#) or [GranularityLevel](#), but not both at the same time. (Some calls may support passing both in the same request. See the “[API Call Reference Guide](#)” on page 765 for information about the filters each call supports.)

If [DetailLevel](#) or [GranularityLevel](#) applies to a call, the definitions of the levels are consistent across all request types (although the list of fields returned varies per call). For example, if [DetailLevel](#) or [GranularityLevel](#) applies to a call:

- Specifying no detail level causes the call to return a brief set of data. This is the default detail level and is recommended for all applicable calls.
- Specifying a granularity level returns more results. You can specify one of three filters: [Coarse](#) (fewer results), [Medium](#), and [Fine](#) (more results).
- A detail level value of [ReturnAll](#) causes the call to return all data, with no filtering. Requesting more data can affect client and server-side performance, so you should use this detail level less often. As the extra data returned using this detail may change less often than other data, you can generally specify this detail level when you first request a particular set of data, and then specify no detail level in subsequent requests for the same data.
- A detail level value of [ItemReturnDescription](#) causes the call to return descriptive information. The exact list of data returned varies depending on the object being returned.

Additional detail level values are also available. See the detail level code list in the schema documentation for a list of the available detail levels.

The [DetailLevel](#) property is defined on the abstract type ([AbstractRequestType](#)). The [GranularityLevel](#) property is defined on a call’s request type (e.g., [GetSellerListRequestType](#)) rather than on the abstract type.

Note that not all calls support detail levels to vary the output. Most calls only support certain detail levels or none:

- Calls that submit data to eBay, such as [AddItem](#), generally do not support varying detail levels, so you do not need to specify the detail level in these requests.
- Several data-retrieval functions, such as [GetSellerList](#) and [GetItem](#), support two or more detail levels.

If you pass a detail level that exists in the schema but that is not valid for a particular request, eBay ignores it and the request is processed normally. Note that you can specify multiple detail levels in a single request, but you can only specify one granularity level in a single request.

To determine whether a particular call supports detail levels and/or granularity levels (and if so, what the effects are), see the chapter for that call in “[API Call Reference Guide](#)” on page 765.

In the SOAP API, this example shows how to specify detail levels in a request. For an example in Java, see the `GetSellerListCall.java` file in the `call` folder of the eBay SDK for Java:

eBay SDK for Java
http://developer.ebay.com/community/featured_projects/?name=eBay+SDK+for+Java

Specifying Unique Identifiers (UUID) for Write Calls

Some API calls that send data to eBay support the use of the [Item.UUID](#) or [InvocationID](#) input parameters. These parameters uniquely identify a given call for a particular application/user combination. The parameters must be exactly 32 characters long, and contain only digits from 0-9 and letters from A-F. The parameters are not case-sensitive.

Use [Item.UUID](#) or [InvocationID](#) with supported calls to prevent the submission of duplicate data. If it is not clear whether or not the call succeeded, such as may happen when there is a network error, you can submit the same data again with the same [Item.UUID](#) or [InvocationID](#). If the previous call did succeed, then the second call will not go through, and the same data will not be submitted twice.

[Item.UUID](#) is recommended for use with calls that write item data. If a duplicate [UUID](#) is used for a call, an error (code 488) will be returned with the item ID of the item previously listed with the same [UUID](#) and a Boolean value indicating whether the duplicate [UUID](#) was sent by the same application.

Use [Item.UUID](#) to uniquely identify the following calls:

- [AddItem](#)
- [RelistItem](#)

If a duplicate **InvocationID** is used for a call, an error (code 21060) will be returned along with the following XML structure:

Example 5-8 Duplicate InvocationID Error Structure

```
<DuplicateInvocationDetails>
  <DuplicateInvocationID>AAAAAAA1111111BBBBBBBB2222222</DuplicateInvocationID>
  <Status>Success</Status>
  <InvocationTrackingID>9202918767</InvocationTrackingID>
</DuplicateInvocationDetails>
```

The **Status** property indicates whether the previous call succeeded (**Success**), or is still in progress (**InProgress**). The **InvocationTrackingID** property provides a call-specific tracking number.

The following calls currently support the use of **InvocationID**:

- [AddOrder](#)
- [AddToltemDescription](#)
- [PlaceOffer](#)
- [ReviseCheckoutStatus](#)
- [Reviseltem](#)

Controlling the Amount of Data Returned

Some responses can potentially contain a large payload. A number of mechanisms are available to help you control the amount of data returned:

- Detail levels, granularity levels, and the output selector – These define subsets of data to return within a call response. For example, a particular detail level might cause the response to include buyer-related data in every result, but not seller-related data. See “[Specifying Detail Levels and Granularity Levels](#)” on page 98 and see “[Selecting Fields to Retrieve](#)” on page 59.
- Time filters – These let you specify that you only want to retrieve records that fall within a particular time window. For example, for [GetSellerList](#), [EndTimeFrom](#) and [EndTimeTo](#) are time filters that limit the response to only those items that end between two specified time stamps. Specifying time filters is like specifying conditions in the WHERE clause of an SQL query.
- Pagination filters – These let you specify the quantity of records that you want to retrieve per call. Each set of records is returned as a page. For calls that support (or require) pagination, you can specify the maximum number of results per page you want. When you retrieve the first page, the response indicates what the total number of pages will be, assuming you would continue to specify the same number of results per page in each subsequent call. Once you have this information, you can determine how many more calls you need to make to retrieve all the data, and you can specify which page you want to retrieve next.
- Business data filters – In some cases, you can also reduce the amount of data returned by restricting the response to only those items that meet other business criteria (e.g., only items that support PayPal as a payment method). The choice of data filters allowed varies per call.

Various topics in this documentation refer to these mechanisms in the context of performing particular tasks. Please see “[Retrieving Items](#)” on page 502 for examples.

Security

The following subsections describe how to make secure API requests.

Before a user on the eBay site can perform any operation of significance (list an item, bid on an item, etc.), that user must log in to the site using an eBay user name and a password. Similarly, for the user of an application to make an API call, it must be determined that application end-user is authorized to do so. Making this determination is called *authentication*. On the eBay site, authentication is accomplished by the user signing in. When an application is acting on a user’s behalf, authentication is accomplished by the application passing authenticating data in the API call’s request. eBay uses the user authentication data passed with the call’s request to identify the requesting user and authenticate the user before performing the requested action.

For many API calls (as is the case with many user activities on the eBay site) this user authentication is merely a verification that the user is a registered eBay user. For instance, anyone who is a registered user may retrieve and view item data with [GetSellerList](#). However, some operations require that the user in question is also verified to be a *particular user*. For example, only an item’s seller is authorized to re-list a particular item. So authentication for an item re-list operation using [RelistItem](#) both verifies that the requester is a registered eBay user and is the item’s seller.

There are two types of authentication data that an application may pass to eBay in a call’s request. One is a user’s authentication token, as shown in the samples in “[Authenticating with an Authentication Token](#)” on page 101.

The other is a pairing of a user’s session ID and the full keyset of your developer credentials, but this is only required for [FetchToken](#), the call that you use to obtain the user’s authentication token in the first place. The full set of developer keys is also required for the [GetTokenStatus](#), [RevokeToken](#), and [GetSessionID](#) API calls. See “[Authenticating with a Full Keyset and Session ID](#)” on page 103.

Authenticating with an Authentication Token

An application can authenticate a requesting user by passing with an API call’s request an authentication token. For more detailed information on getting and renewing authentication tokens, see “[Getting Tokens](#)” on page 117.

In the SOAP API, to authenticate the requester for a call using an authentication token, pass the token in the header for the request. The example below shows the header element that contains the token, as well as the Developers Program keys:

Example 5-9 Passing Token Authentication (SOAP)

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" 
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
    <soap:Header>
        <ebl:RequesterCredentials soapenv:mustUnderstand="0">
            <eBayAuthToken>
                ... AUTHENTICATION TOKEN GOES HERE ...
            </eBayAuthToken>
        </ebl:RequesterCredentials>
    </soap:Header>
    ... Call body ...
</soap:Envelope>
```

In the XML API, to authenticate the requester for a call using an authentication token, pass the token in the `eBayAuthToken` element within the `RequesterCredentials` element in the request XML.:

Example 5-10 Passing Token Authentication (XML)

```
<?xml version="1.0" encoding="utf-8"?>
<GeteBayOfficialTimeRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <RequesterCredentials>
        <eBayAuthToken> Token goes here </eBayAuthToken>
    </RequesterCredentials>
    <Version>383</Version>
</GeteBayOfficialTimeRequest>
```

A sample delivered with the eBay SDK for Java shows how to add this element in Java and populate it with an authentication token. Specifically, see the `Global.java` file in the `signingredirect` folder in the eBay SDK for Java:

eBay SDK for Java
<http://developer.ebay.com/developercenter/java/sdk/>

Example 5-11, which applies to the SOAP API, shows how to add this header child element in C# and populate it with an authentication token.

Example 5-11 Passing Token Authentication (C#)

```
service.RequesterCredentials = new CustomSecurityHeaderType();

// Specify the authentication token for the requesting user
service.RequesterCredentials.eBayAuthToken = UserToken;
```

If you are using the eBay SDK for Java, pass the authentication token for the requesting user in the `ApiCredential` property of the `ApiClient` object used for the call. The `ApiClient.ApiCredential.eBayToken` property holds the authentication token for a call. The

authentication token is conveyed in the variable `token`, which is of type `string`. For an example in Java, see the `Global.java` file in the `signingredirect` folder of the eBay SDK for Java:

eBay SDK for .NET
<http://developer.ebay.com/developercenter/windows/sdk/>

In the eBay SDK for .NET, pass the authentication token for the requesting user in the `ApiCredential` property of the `ApiClient` object used for the call. The `ApiClient.ApiCredential.eBayToken` property holds the authentication token for a call. In the example below, the authentication token is conveyed in the variable `token`, which is of type `string`.

Example 5-12 Passing Token Authentication (C#)

```
// Create the ApiClient object and specify the credentials
ApiClient context = new ApiClient();
context.ApiCredential.ApiAccount.Application = "YourAppId";
context.ApiCredential.ApiAccount.Certificate = "Your";
context.ApiCredential.ApiAccount.Developer = "YourDevID";
context.ApiCredential.eBayToken = "ABC...123";
```

Authenticating with a Full Keyset and Session ID

The use of a full application keyset (`DevID`, `AppID`, and `AuthCert`) to authenticate the requester of a call is not required for most calls.

However, a few calls do require the full keyset. These exceptional calls are all associated with authentication and tokens: `FetchToken`, `RevokeToken`, `GetTokenStatus`, and `GetSessionID`.

For these calls, you pass the application's full keyset along with the request.

For SOAP and XML, you pass the keyset in the header for the request. The `FetchToken` examples for SOAP and XML below show header elements that contain the keyset-based authentication data:

Example 5-13 Passing the Keyset in the Header (SOAP)

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" 
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ebc="urn:ebay:apis:eBLBaseComponents">
    <soap:Header>
        <ebc:RequesterCredentials soapenv:mustUnderstand="0">
            <Credentials>
                <DevId>MyDevId</DevId>
                <AppId>MyAppId</AppId>
                <AuthCert>MyCertId</AuthCert>
            </Credentials>
        </ebc:RequesterCredentials>
    </soap:Header>
    <FetchTokenRequest xmlns="urn:ebay:apis:eBLBaseComponents">
        <SessionID>MySessionID</SessionID>
        <Version>613</Version>
    </FetchTokenRequest>
</soap:Envelope>
```

Example 5-14 Passing the Keyset in the Header Element (XML)

```
Headers:  
X-EBAY-API-COMPATIBILITY-LEVEL: 613  
X-EBAY-API-DEV-NAME:YourDevID  
X-EBAY-API-APP-NAME:YourAppID  
X-EBAY-API-CERT-NAME:YourCertID  
X-EBAY-API-CALL-NAME:FetchToken  
X-EBAY-API-SITEID:0  
Content-Type:text/xml  
  
Request Payload:  
<?xml version="1.0" encoding="utf-8"?>  
<FetchTokenRequest xmlns="urn:ebay:apis:eBLBaseComponents">  
    <Version>613</Version>  
    <SessionID>MySessionID</SessionID>  
</FetchTokenRequest>
```

A sample delivered with the eBay SDK for Java shows how to pass authentication data based on your eBay Developer's Program credentials in Java. Specifically, see the `ClientAuthenticationHandler.java` file in the `handler` folder of the eBay SDK for Java:

eBay SDK for Java
<http://developer.ebay.com/developercenter/java/sdk/>

If you are using the eBay SDK for Java, pass your eBay Developer's Program credentials within an API call request in the `ApiCredential` property of the `ApiClient` object. The `ApiCredential` object has an `ApiAccount` property to contain the application's credentials and an `eBayAccount` property to hold the requesting user's eBay Developer's Program credentials. These are used to authenticate the user for the call. Variables are used for the application's credentials (`mDevID`, `mAppID`, and `mCert`) and for the eBay Developer's Program credentials. The variables are all of type `String`.

eBay SDK for .NET
<http://developer.ebay.com/developercenter/windows/sdk/>

If you are using the eBay SDK for .NET, pass your eBay Developer's Program credentials within an API call request in the `ApiCredential` property of the `ApiClient` object. The `ApiCredential` object has an `ApiAccount` property to contain the application's credentials and an `eBayAccount` property to hold the requesting user's eBay Developer's Program credentials, which are used to authenticate the user for the call. The variables used to express the application's credentials (`devID`, `appID`, and `cert`) and the variables containing the eBay Developer's Program credentials are all of type `string`.

Standard Output Data

All API calls return certain fields, and some calls may return additional call-specific data that you requested.

The standard fields are defined on [AbstractRequestType](#). For or information on the call-specific input tags, see the eBay Trading API Call Reference. The output data common to all calls is described in the next few sections.

- Time stamp
- Version
- Build
- Acknowledgement
- Correlation ID
- Message (not to be confused with the MessageID input field)
- Hard expiration warning
- Errors

Time Stamp

All responses return the official eBay time stamp in UTC/GMT (see “[eBay Types](#)” on page 959 for information about UTC/GMT values).

The time stamp indicates the time when eBay processed the request; it does not necessarily indicate the current eBay official eBay time. In particular, calls like [GetCategories](#) can return a cached response, so the time stamp may not be current.

Version and Build

Use the [Version](#) and [Build](#) fields to determine the version of the schema and the specific software build that eBay used when processing the request and generating the response. In particular, Developer Technical Support may ask you for the [Build](#) value when you work with them to troubleshoot issues.

See “[Release Versions](#)” on page 84 for details about working with different schema versions.

Acknowledgement

All responses return a standard [Ack](#) field that indicates the success, failure, or partial failure of a call.

Some calls that you use to submit data or intructions to eBay return an [Ack](#) status of [Success](#), and no call-specific data. For example, when you use [CompleteSale](#) to change an item’s paid or shipped status in My eBay, all you need to know is that your request succeeded.

Other data-submission calls return [Success](#) and include other fields that you need in order to perform subsequent tasks. For example, when you use [AddItem](#) to list an item, it returns the item ID, the start and end times, and other useful data.

When a call is successful, it can also return useful warnings. For example, a warning could indicate that your authentication token is about to expire, you passed in an unrecognized field, some data in your request was dropped, or other useful information. Please see “[Error Handling](#)” on page 107 for more information about working with warnings.

If the **Ack** field returns **Failure**, it means your request could not be processed. In this case, one or more errors are returned to explain the cause of the failure, and no other call-specific data is returned. Please see “[Error Handling](#)” on page 107 for more information about handling errors.

A very limited number of calls can return an **Ack** status of **PartialFailure**. This means some portion of your request succeeded and another portion failed. See [CompleteSale](#) for an example.

Correlation ID

If you pass a **MessageID** in a request, we will return the same value as the **CorrelationID** in the response. This may be useful for tracking that a response is returned for every request and to match particular responses to particular requests. See “[Specifying a Message ID to Correlate the Request and Response](#)” on page 96 for more information.

Message

The **AddItem** family of calls ([VerifyAddItem](#), etc.), **EndItem**, **PlaceOffer**, and certain other calls can return a **Message** field that provides the seller with information critical to the listing’s success. This can include listing hints, policy violation explanations, or other details. (See the **Field Index** in the *eBay Trading API Call Reference* for a complete list of calls that return this field.)

Applications must recognize whenever the **Message** field is returned and provide a means to display the message to the user.

These messages provide these key benefits to the end user:

- Sellers can review and resolve potential policy violations to reduce the likelihood that their item will be ended administratively by eBay.
- Sellers can take advantage of the hints as tools to increase their sales. For example, a hint could advise newer sellers that including pictures has been shown to increase the success of listings by a certain percentage.

For listing use cases, we strongly recommend that you use [VerifyAddItem](#) to surface these messages to sellers before they submit their listings. (You should still check the [AddItem](#) response for any additional messages.) By ensuring that sellers receive policy messaging and other information in a timely manner, you can help to improve a seller’s experience on eBay. Neglecting to surface this information may lead to items being ended and general seller dissatisfaction.

When errors are returned, the **Message** field may be surfaced for both blocking (failure) and warning scenarios (in addition to the **Errors.LongMessage** and **Errors.ShortMessage** fields). In this case, the **Message** field typically contains a more detailed explanation of why the error occurred and how to fix the problem.

Important: If a request succeeds with a warning instead of failing with a severe error, this doesn’t mean the message is less important. For example, if a listing appears to violate a zero-tolerance policy, but the certainty is less than 100% at the time of submission, the listing request may succeed with a warning. In this case, the **Message** field might return a policy message with guidance and education related to the issue. If the seller doesn’t revise the listing with the required changes soon, eBay may still administratively end the listing a short time later due to the policy violation.

The **Message** value may return HTML, such as <TABLE> and tags and URLs. If your application doesn't support HTML, you can parse the tags and then translate the data into UI elements particular to the programming language you use.

Errors and Warnings

For any request, one or more errors or warnings may be returned. Please see “[Error Handling](#)” on page 107 for more information about working with errors.

Hard Expiration Warning

If the user's authentication token is about to expire, all calls using that token will return a **HardExpirationWarning** field with the expiration date. See “[Hard Expiration Warning](#)” on page 107.

Error Handling

As with all applications, your application should implement at least basic error handling. Doing so grants a number of benefits to the application and the application's end-users.

The most prevalent benefit is the graceful recovery of the application from error states. Without even the simplest form of error handling in an application, its execution can be brought to a complete halt by an error returned by eBay.

Another benefit is that the application can display errors that are more meaningful to the end-user, who will likely not know about the errors that could occur.

Also, by incorporating error handling, an application can more easily be localized for use in different languages. If the application's error messages that correspond to the API error messages are stored in something like a string table, then different string tables can readily be substituted for different languages.

When a problem occurs and a request (or a portion of a request) cannot be executed for some reason, the response includes a list of errors instead of the normal business data. If the request is executed successfully but a minor problem was found or a change was made that you might not expect, the response includes the normal business data accompanied by a list of warnings.

It is a good idea to understand the types of errors and warnings that eBay may return so that you can design your application to handle them gracefully. See also “[Listing Policies](#)” on page 174.

Also, sometimes errors may occur due to problems on eBay's side. For example, error 10007 (“Internal error to the application”) indicates an error on the eBay server side, not an error in your application. It is important to understand how to determine whether this has occurred, and when to contact eBay for help.

[“Error Types” on page 108](#)

[“Customizing Your Application's Error Handling Abilities” on page 113](#)

[“Compatible Application Check Requirements for Error Handling” on page 115](#)

[“Getting Help When You Cannot Identify the Source of an Error” on page 115](#)

Error Types

The eBay Platform defines two general types of errors:

- Application-level errors

These occur due to problems with business-level data. In these cases, we return a list of errors in the response message payload. See “[Application-Level Errors](#)” on page 108 for details.

- Infrastructure errors

In the SOAP API, these indicate that an infrastructure error has occurred, such as a problem on eBay’s side with a database or server going down, or a problem with the client-side or server-side SOAP framework. When infrastructure error occurs, we return a list of SOAP faults. See “[Infrastructure Errors](#)” on page 111 for details.

Please see “Error Message Strings” for the latest list of possible error codes and messages that can be returned. If you are reading the PDF version of this guide, you can access the error message strings documentation via the online version of this guide. See the online version of this topic (or the online version of “[How this Documentation is Organized](#)” on page 40) for a link to the Error Message Strings documentation.

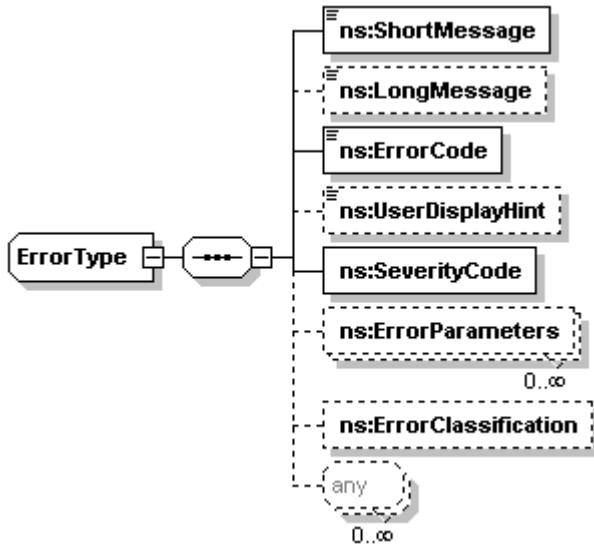
Application-Level Errors

Application-level errors occur due to problems with business-level data on the client side or on the server side. For example, these are just a few situations that could cause application-level errors:

- The request contains an invalid combination of fields (e.g., a reserve price for a Fixed Price listing).
- The request is missing a required field.
- The request is valid, but a problem occurred in eBay’s internal business logic while processing the data.

When an application-level error occurs, eBay returns an error object ([Error](#)) in the response. Application-level errors are returned as normal HTTP 200 responses. Each error includes a unique code ([ErrorCode](#)) that identifies the particular error condition that occurred. It also provides messages ([ShortMessage](#) and [LongMessage](#)) that indicate the cause of the problem. The text of each message also implies whether the error was due to a problem with the request (e.g., a field was invalid) or due to a problem on the server.

Figure 5-4 The API Error Model



If you are using the eBay SDK for Java or the eBay SDK for .NET, please note that the SDK translates these SOAP application-level errors into `ApiException` objects. These are returned in the response object for an API call. Applications access exception objects for a particular call through the `ApiException` property of the concrete API call class that was used when the error(s) occurred (such as `AddItemCall.ApiException` if the errors occur when listing a new item). Concrete API call classes “inherit” the `apiException` property from the `ApiCall` class. At run-time, an application can use the `hasError()` and `hasWarning()` methods of the API call object to determine whether an error (or a warning) has occurred as a result of invoking that object’s execution method (e.g., the `addItem()` method for an `AddItemCall` object). If an error (or a warning) is indicated, then the application accesses the exceptions conveyed in the object’s `ApiException` property.

If you are using the eBay SDK for Java or the eBay SDK for .NET, each `ApiException` object can convey multiple exceptions. The `ApiException` object is an array of objects of type `ErrorType`. Properties of an `ErrorType` object include:

Table 5-4 Key `ErrorType` Properties

Property	Meaning
<code>ErrorClassification</code>	One of two classes: system errors and request errors.
<code>ErrorCode</code>	API code that uniquely identifies the error.
<code>ErrorParameters</code>	Context-specific error variables that indicate details about the error condition. These are useful when multiple instances of <code>ErrorType</code> are returned.
<code>LongMessage</code>	Verbose error message.
<code>SeverityCode</code>	Code indicating the severity of the error that occurred. Will be a value of <code>Error</code> or <code>Warning</code> .
<code>ShortMessage</code>	Brief error message.
<code>UserDisplayHint</code>	Indicates whether the error is meant to be parsed by the application, or displayed directly to the end user.

An error can have one of two severity levels. The severity level indicates whether the request should be resent or another action needs to be taken in response to the error.

Table 5-5 Error Severity Levels

Severity Level	Meaning
Error	<p>This means the request that triggered the error was not processed successfully.</p> <p>When a serious application-level error occurs, the error is returned instead of the business data.</p> <ul style="list-style-type: none"> ■ If the source of the problem is within the application (such as a missing required element), please change the application before you retry the request. If the problem is due to end-user input data, please alert the end-user to the problem and provide the means for them to correct the data. Once the problem in the application or data is resolved, you can attempt to re-send the request to eBay. ■ If the source of the problem is on eBay's side, An application can retry the request as-is a reasonable number of times (eBay recommends twice). If the error persists, contact Developer Technical Support. Once the problem has been resolved, the request may be resent in its original form. <p>Please see "Compatible Application Check" on page 67 for more information.</p>
Warning	<p>This means the request was processed successfully, but something occurred that may affect your application or the user. For example, eBay may have changed a value the user sent in. In this case, eBay returns a normal, successful response and also returns the warning.</p> <p>When a warning occurs, the error is returned in addition to the business data. In this case, you do not need to retry the request (as the original request was successful). However, depending on the cause or nature of the warning, you might need to contact either the end user or eBay to effect a long term solution to the problem to prevent it from reoccurring in the future.</p>

Some application-level error messages return additional variables that contain specific information about the error context. For example, a theoretical error might look like this:

```
Attribute Set Id {0} does not match category you have entered for this request {1}. Please refresh the meta-data.
```

The variables are identified by error parameter ([ErrorParameter](#)) objects. If you are using an SDK, they are returned in the [ErrorType.ErrorParameters](#) property in the form of an [ErrorParameterType\[\]](#) list. These are intended to help you distinguish the context of errors more easily when a response contains multiple errors. Applications can also use these parameters to more easily flag fields that users need to correct.

Each error can contain a list of 0 or more parameters. Each one has a parameter ID attribute ([ParamID](#)) that specifies the index of the parameter in the list (e.g., 0 for the first parameter, 1 for the second parameter). It also specifies a value ([Value](#)), which is the parameter itself (e.g., the attribute set ID).

The following sample, made using the XML API, generated both an error and a warning.

Example 5-15 Sample XML call input leading to error and warning

```
<?xml version="1.0" encoding="utf-8"?>
<GetBestOffersRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <Version>304</Version>
  <ItemID>123456</ItemID>
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
</GetBestOffersRequest>
```

Example 5-16 Sample XML call output containing error and warning

```
<?xml version="1.0" encoding="utf-8"?>
<GetBestOffersResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2006-05-31T02:34:59.381Z</Timestamp>
  <Ack>Failure</Ack>
  <Errors>
    <ShortMessage>Header version 445 does not match request version 304.</ShortMessage>
    <LongMessage>The version 445 in the HTTP header X-EBAY-API-COMPATIBILITY-LEVEL does not
match the version 304 in the request. The HTTP header version will be used.</LongMessage>
    <ErrorCode>21926</ErrorCode>
    <SeverityCode>Warning</SeverityCode>
    <ErrorParameters ParamID="0">
      <Value>445</Value>
    </ErrorParameters>
    <ErrorParameters ParamID="1">
      <Value>304</Value>
    </ErrorParameters>
    <ErrorClassification>RequestError</ErrorClassification>
  </Errors>
  <Errors>
    <ShortMessage>Item not found.</ShortMessage>
    <LongMessage>Item "123456" is invalid, not activated, or no longer in our database.</
LongMessage>
    <ErrorCode>21549</ErrorCode>
    <SeverityCode>Error</SeverityCode>
    <ErrorParameters ParamID="0">
      <Value>123456</Value>
    </ErrorParameters>
    <ErrorClassification>RequestError</ErrorClassification>
  </Errors>
  <Version>463</Version>
  <Build>e463_core_Bundled_3001562_R1</Build>
</GetBestOffersResponse>
```

eBay SDK for Java
http://developer.ebay.com/community/featured_projects/?name=eBay+SDK+for+Java

Infrastructure Errors

In the SOAP API, an infrastructure error indicates a communication failure on the part of either your application (including your SOAP client) or the eBay servers. This might include such things as passing an invalid token to authenticate the API call's requesting user.

When an infrastructure error occurs, eBay returns a SOAP fault that specifies the details of the errors (**FaultDetail**). Note that in certain cases a SOAP fault generated by the web services framework (AXIS) may be returned without a **FaultDetail** node included. An example that can trigger such a case is if an integer is specified for the **DetailLevel** in the request instead of a valid value for that property. SOAP faults are returned as HTTP 500 error responses, in accordance with the SOAP 1.1 standard.

The fault details object includes a unique code (**ErrorCode**) that identifies the particular error condition that occurred. It also provides a message (**DetailedMessage**) that indicates the cause of the problem.

If you are using the eBay SDK for Java, this error classification includes AxisFault exceptions.

For the SOAP API, four general types of SOAP faults are used for infrastructure errors.

Table 5-6 Infrastructure Error Types

Fault Type	Meaning
Client faults	<p>These indicate that a problem occurred with the meta-data that your application submitted. These types of faults fall into these categories:</p> <ul style="list-style-type: none"> ■ Request authentication failures (unsupported verb, the application is not allowed to make this request, etc.) ■ Application authentication failures (invalid certificates, invalid or missing tokens, etc.) ■ User authentication failures (the requesting user is not a registered member, etc.) ■ Malformed requests (the XML is not well-formed, the encoding is not supported, etc.)
Server faults	<p>These indicate that a problem occurred within eBay's infrastructure. These are some examples of conditions that result in such faults:</p> <ul style="list-style-type: none"> ■ The API servers are not available. ■ A database connection error occurred.
VersionMismatch fault	<p>These indicate that the SOAP envelope information your application passed is incorrect. Be sure the localname and namespace match the SOAP version we support, which is SOAP 1.1.</p>
MustUnderstand fault	<p>These indicate that a problem occurred with the Web services infrastructure. For example, this type of fault may occur when the SOAP processor on the server cannot process a header element that was sent via your SOAP client. If you are using one of a SOAP client (see "Supported Client Environments" on page 50), please report the issue to eBay Developer Support. If you are using a different SOAP client, please see the Release Notes or Known Issues before contacting eBay to discuss the problem.</p>

The SDKs translate SOAP infrastructure-level errors (and, in the case of the eBay SDK for Java, AxisFault errors) into [ApiException](#) and [SdkException](#) objects. Two key properties are the following: the [ErrorType](#) property (indicating the type of error) of [ApiException](#) and the [Message](#) property (containing the text message of the exception) of [SdkException](#).

An infrastructure error can have one of two severity levels. The severity level indicates whether the request should be resent or another action needs to be taken in response to the error.

Table 5-7 Infrastructure Error Severity Levels

Severity Level	Meaning
Error	<p>This means the request was not processed.</p> <ul style="list-style-type: none"> ■ If a serious infrastructure error occurs due to a problem on the client side, please resolve the problem before attempting to retry the request. ■ If a serious infrastructure error occurs due to a problem on eBay's side, your application can retry the request as-is a reasonable number of times (eBay recommends twice). If the error persists, contact Developer Technical Support. Once the problem has been resolved, your application can re-send the request in its original form. <p>Please see "Compatible Application Check" on page 67 for more information.</p>
Warning	<p>This means the request was processed successfully, but something occurred that you should be aware of.</p> <ul style="list-style-type: none"> ■ If a client-side warning is returned, you do not need to retry the request. However, depending on the cause or nature of the warning, you may want to resolve the problem. ■ If a server-related warning is returned, you do not need to retry the request. However, you may want to contact Developer Support to effect a long term solution to the problem to prevent it from reoccurring in the future.

The following example, which applies to the SOAP API, shows one way to handle infrastructure error message data using C#.

Example 5-17 Catching Infrastructure Errors (C#)

```
System.Xml.XmlNode details = soapex.Detail;

string severity = "";
string longmsg = "";
string errcode = "";
string srtmsg = "";

try
{
    severity = details.SelectSingleNode("/FaultDetail/Severity").InnerText;
    longmsg = details.SelectSingleNode("/FaultDetail/DetailedMessage").InnerText;
    errcode = details.SelectSingleNode("/FaultDetail/ErrorCode").InnerText;
    srtmsg = soapex.Message;
}
catch (System.NullReferenceException)
{
    severity = "Error";
    longmsg = soapex.Message;
    errcode = "0";
    srtmsg = soapex.Code.Name;
}
SeverityCodeType type = SeverityCodeType.Error;
if (severity != "Error")
    type = SeverityCodeType.Warning;

errorType.ErrorCode = errcode;
errorType.SeverityCode = type;
errorType.LongMessage = longmsg;
errorType.ShortMessage = srtmsg;
// Return or print the error data you need
```

Other Errors in the SDKs

In the case of the SDKs, all other exception types (aside from application-level and infrastructure-level errors) are translated into [SdkException](#) objects. Surround critical application operations in a `try..catch` construct to handle these exceptions. Use the [SdkException.getInnerThrowable\(\)](#) method to get the exact exception that occurred.

Customizing Your Application's Error Handling Abilities

An application can use any combination of the error properties for its own error handling. For example, an application might have a specialized module for handling error states. When such a state is detected, the error code could be passed to the error-handling module and used in a decision branch construct to take a special action unique to a given error condition.

To return error messages in a language other than English, specify an appropriate value in the [ErrorLanguage](#) field of the request.

To check whether you have accidentally misspelled a field name, passed in a field that does not exist, or if you are using deprecated input fields, set [WarningLevel](#) to a value of [High](#). We recommend that you only use this for testing and debugging purposes, not in your production application. We have three error messages associated with [WarningLevel](#).

Table 5-8 WarningLevel Messages

Error Code	Long Error Message	Meaning
21927	The element {0} was found in the input SOAP request. This is not a declared element in the schema and will be ignored.	This means you have specified an element that we don't recognize. (This could occur if you spell the element name differently than we do, or if you send it in the wrong node.)
219450	The input object "{0}" is deprecated as of schema version {1}. Replacement: "{2}".	This means the object is deprecated as of the specified schema version, and we will stop supporting it for all versions in the future.
219451	The input object {0} is no longer supported and will be ignored.	This means we no longer support the object for any schema version.

Please see ["Error Message Strings"](#) on page 966 for the latest list of all possible error codes and messages that can be returned. If you are reading the PDF version of this guide, you can access the error message strings documentation via the online version of this guide. See the online version of this topic (or the online version of ["Error Message Strings"](#) on page 966) for a link to the Error Message Strings documentation.

If you are using the SDKs, errors come from two sources. API errors are raised by eBay (see ["Application-Level Errors"](#) on page 108 and ["Infrastructure Errors"](#) on page 111). Errors also are raised that are Java or .NET application errors, or errors that might occur as a result of attempting to use SDK classes. These errors need to be handled in different ways.

To handle Java or .NET errors, enclose important and volatile operations (such as creating an object and accessing its methods) in a `try..catch` construct.

If you are using the SDKs, API errors will not trigger an exception that will be captured by a `try..catch` construct. Rather, the API call object's execution method would be processed normally, but instead of a result set consisting of data there will be an error construct returned (or data and a warning type error construct in the case of warnings). An API error of this sort typically occurs due to bad or invalid data being supplied as input or by a required value not being supplied. For example, if [AddItemCall](#) is used to list an item and no starting price is specified, an API error would result.

If you are using the SDKs, determine if there are any API errors or warnings after a call by calling the [hasError\(\)](#) and the [hasWarning\(\)](#) methods of the API call class ([AddItemCall](#), [GetSellerListCall](#), etc.). The [hasError\(\)](#) method returns a value of true if any API exceptions resulted from the call. In this case, get the call object's exception object using the [getApiException\(\)](#) method and inspect the exceptions it contains to determine what went wrong. If an API exception occurs, the call results in the return of just an error and no data is returned. The [hasWarning\(\)](#) method returns a value of true if a warning was returned as a result of the call. Visit the [ApiException](#) property of the call object to retrieve the warning. When warnings occur, the call will have been processed normally (such as the listing of an item) and the call returns the normal result set. It is just that the normal result set is also accompanied by the warning.

Compatible Application Check Requirements for Error Handling

We offer a general set of guidelines that you should follow to determine what actions to take to correct errors and when to retry a request. Please see “[Compatible Application Check](#)” on page 67 for these guidelines.

Getting Help When You Cannot Identify the Source of an Error

If you have trouble determining the cause of an error, please see “[Feedback and Support](#)” on page 35 for a list of resources that are available to you.

Please note that error 10007 (“Internal error to the application”) indicates an error on the server side, not an error in your application.

The Object Model of the eBay SDK for Java

This section is relevant if you are interested in using the eBay SDK for Java to interact with the Trading API. The classes of the eBay SDK for Java are divided into these main libraries:

- SDK Kernel Library
- API Call Library
- Attributes Library

Additional helper classes and utilities are also available.

For more information, see the javadoc at the following location:

<http://developer.ebay.com/DevZone/JavaSDK/docs/LibRef/index.html>

eBay SDK for Java Libraries

The classes of the SDK Kernel Library define the basic functionality that is common to and used by all API call classes. Each API call class extends the [ApiCall](#) class in this library, “inheriting” the basic functionality defined in [ApiCall](#) and adding functionality that is specific to the particular API call the API call class encapsulates. Also defined in this library is the [ApiClientContext](#) class, which is used to set up the execution environment (or *API context*) for an API call.

The classes of the Picture Service Library define the functionality needed for an application to use item picture files hosted on the eBay Picture Services (EPS) server. These include the classes [eBayPictureService](#) and [PictureInfo](#).

The API Call Library defines the classes that encapsulate individual API calls. API call classes “inherit” basic functionality from [ApiCall](#), like: authentication constructs, target server URLs, exception container constructs, and request and response objects. While [ApiCall](#) implements abstract request and response objects, the individual API call classes define the concrete request and response objects that are needed to perform the operations associated with the API call each wraps. In addition to the functionality an API call class “inherits” from [ApiCall](#), it also defines functionality specific to the API call represented by the call class. For instance, [AddItemCall](#) defines the

functionality to send an [Item](#) object to eBay using the [AddItem](#) API call and then return the estimated listing fees for that item as the return value of the execution method for the [AddItemCall](#) class.

Call Class Basics in the eBay SDK for Java

API call classes are the front-facing aspect of the eBay SDK for Java. Applications use these classes more than any other. API call classes are the means for an application to execute an API call, passing data to eBay or retrieving data from eBay.

Each API call class is named for the API call it encapsulates. The name of the API call class is the name of the API call plus the word “Call” appended to the end. For instance, the SOAP API call [GetSellerList](#) is represented in the eBay SDK for Java by the API call class named [GetSellerListCall](#).

As is the case for the API calls the API call classes represent, API call classes fall into two general categories: those that send data to eBay and those that retrieve data from eBay. In general, call classes that send data to eBay have names that start with “Add,” “Set,” “Relist,” and “Revise.” The [AddItemCall](#) and [RelistItemCall](#) classes are examples. Generally speaking, the input data sent to eBay by these classes tends to be comparatively larger than the data they get back from eBay. Typically, the data coming back to the application for these classes is related to confirmation of the operation. Call classes that retrieve data from eBay are typified by names that start with “Get,” like [GetSellerList](#), [GetItem](#), and [GetUser](#). For these call classes, the data sent is typically small compared to the amount of data retrieved when the call is made. For example, with [GetItemCall](#) only an item ID is sent to eBay, but the data for an entire item listing is returned (in an [Item](#) object).

Each API call class has one execution method by which the application triggers the API call. The execution method for a particular API call class is named for the corresponding API call, but with a lowercase letter at the beginning. For example, the execution method for the [AddItemCall](#) class is named [addItem\(\)](#), for the [AddItem](#) API call it invokes.

Applications follow a common design pattern for every API call class:

- 1 Create and set up an [ApiClient](#) object that defines the call’s environment. This object is used for such things as specifying the URL for the target eBay API server and specifying the authentication data for the end-user making the call.
- 2 If the call class is one that sends data to eBay, create an API schema object that is of a type appropriate for the operation and the API call class used. For example, an [Item](#) object is needed for listing an item with [AddItemCall](#).
- 3 Create the API call object and call its execution method.

If the call class is one that sends data to eBay, the application will either assign the schema object to one of its properties or pass it as the execution method’s input. If the call is one that retrieves data from eBay, the application will receive the retrieved data either as the return value of the execution method or in properties of the call object’s response object.

Chapter 6

Getting Tokens

Your application needs user tokens to make authenticated and authorized eBay API calls on behalf of your users. For a thorough definition of tokens, see ["What is a Token?"](#) on page 127.

This chapter describes how to configure your application to get tokens for users, how to get tokens that will enable users to make use of your application, how to use tokens, and how to replace them when they expire.

For a brief demonstration of how a web application gets a token for a user, see the API flow tutorial "Getting Tokens":

<http://developer.ebay.com/DevZone/xml/docs/HowTo/Tokens/GettingTokens.html>

In the following sections it is assumed that you have joined the eBay Developers Program and have a keyset. If you haven't done this yet, you can do so on the [Quickstart page](#).

- ["Single-User Applications" on page 117](#)
- ["Setting Up an Application to Receive Tokens" on page 118](#)
- ["Getting Tokens for Applications with Multiple Users" on page 121](#)
- ["Using Tokens" on page 127](#)
- ["Watching for Token Expiration" on page 128](#)
- ["About Tokens" on page 127](#)

Single-User Applications

If your application is for personal use and only needs to support one user, you only need one token. You can generate a single token for your application's user using the token tool.

Token Tool
<http://developer.ebay.com/tokentool/>

The token user must be distinct from your own developer username and login. You can read about creating test users in ["Creating a Test User"](#) on page 63.

Configuring the Consent Flow for Applications with Multiple Users

This information has been updated and moved to ["Setting Up an Application to Receive Tokens"](#) on page 118.

Setting Up an Application to Receive Tokens

This section explains how to specify token-related preferences for your application and set up your application's consent form.

In the consent form, users agree to grant your application access to their eBay data. eBay supplies this permission to you in the form of a token that your application includes in API calls it makes on the user's behalf.

A consent form as a user sees it looks like this:



In this section:

- “Set Application-Level Settings” on page 118
- “Generate an RuName for Your Application” on page 119
- “Set RuName-Level Settings” on page 120

Set Application-Level Settings

These settings customize your application's consent form with your logo and a URL of your choice.

- 1 Log on to your [MyAccount page](#).
- 2 Select the “application settings” tab.
- 3 Click “Customize the eBay User Consent Form.”
- 4 Select the environment and the keyset for your application.

Use the Sandbox environment for testing, and the Production environment to set up your application for eBay users.

- 5 Show Application Detail: enable this setting so that your application's consent form will display your logo, URL, and description.

- 6 Application URL: specify the URL you want to have displayed in the consent form, usually an information page about your application.
- 7 Application Logo: uploads and formats your logo.

Acceptable formats for application logos include JPG, GIF, PNG, BMP, and TIF image files. JPG is recommended. If you use PNG, it will be converted to JPG (or GIF) format. The maximum file size is 7MB. eBay Picture Services (EPS) downscales and compresses the picture to store it at the different sizes in the imageset. For best results, upload a picture that has a minimum of 1000 pixels on the longer side.

Generate an RuName for Your Application

Your application's RuName identifies your application to eBay when you send users to the eBay sign-in page and the user consent form.

The RuName is also associated with settings that control how your application interacts with prospective users. You can specify multiple RuNames for your application, if localization or marketing considerations make it desirable for your application to have multiple display titles and other settings.

To generate an RuName:

- 1 Make sure you are still in the “application settings” tab in your [MyAccount page](#), with your environment and keyset still selected.
- 2 Under “Manage Your RuNames,” click “generate runame.”

The generated RuName appears under “Manage Your RuNames.”

Set RuName-Level Settings

RuName-level settings determine how your application interacts with a user who wants to grant a token, specifically:

- The content that users will see on the consent form
 - Rules about how the application authenticates users and gets tokens
 - Application URLs, such as the AcceptURL that eBay uses to return consenting users to the application
- 1 Under Manage Your RuNames, click “Show Details.”

Display Title	<input type="text" value="MyeBayApp"/>
Display Description	<input type="text" value="An eBay application."/>
Token Return Method	<input type="text" value="FetchToken"/>
Authorization Type	<input type="text" value="Authorization"/>
Application Type	<input type="text" value="Web Based"/>
Accept Redirect URL	<input type="text" value="https://arribada.ebay.com/aw-secure/a"/>
Reject Redirect URL	<input type="text" value="https://arribada.ebay.com/aw-secure/a"/>
Privacy Policy URL	<input type="text"/>

- 2 Display Title: enter the company or application name to be displayed on the consent form.
- 3 Display Description: enter text describing your application to be displayed on the consent form.
- 4 Token Return Method: Select “FetchToken,” and your application will use the **FetchToken** to retrieve the token once the user has consented. “FetchToken” is the recommended method. If your current setting reads “Redirect,” please plan to move to using the FetchToken method.
- 5 Authorization Type: there are two types: Authorization and IDVerification. If your application needs to access eBay users’ data via eBay Web Services, set the Authorization Type to Authorization.

Set Authorization Type to IDVerification if your application needs only to confirm users’ eBay UserIDs, and does not need to access users’ private data. Users will be presented with a more casual text on the consent form, and a token is not generated for the application. The application needs to use the **ConfirmIdentity** API call to complete the final UserID confirmation step. Because there is no token, there is no need to call **FetchToken**.

- 6 Application Type: specify whether your application is web-based and has a URL to which you would like your users returned after they consent, or is a desktop application without a URL.

- 7 Accept Redirect URL: the URL to which users will be directed after they consent to the web application authenticating them. This URL must support SSL and must use the HTTPS protocol.
If your application is capable of serving web pages, you *should* provide your own web page and set this URL. If your application cannot serve web pages, this URL defaults to a standard eBay accept-response page (the eBay page and URL are subject to change by eBay).
- 8 Reject Redirect URL: the URL to which users are directed when they do not consent.
If your application is capable of serving web pages, you *should* provide your own web page and set this URL. If your application cannot serve web pages, this URL defaults to a standard eBay accept-response page (the eBay page and URL are subject to change by eBay).
- 9 PrivacyPolicyURL: specify the URL at which your application's privacy policy can be read.
- 10 Click "Save Settings."

Your application is now ready to get tokens.

Getting Tokens for Applications with Multiple Users

With one-time setup complete (see "[Setting Up an Application to Receive Tokens](#)" on page 118), your application can respond to prospective users and get tokens for them.

Note: If there is only one user of the application, a much simpler method is available for getting tokens. See "[Token Tool](#)" on page 117.

Here is a summary of the process that results in your web application getting a token for a user:

- 1 A user indicates a desire to use your application.
- 2 Your application requests a session ID from eBay, and receives one.
- 3 Your application constructs a URL that contains the session ID and an identifier for your application, and uses this URL to send the user to the eBay sign-in page.
- 4 The user signs in to eBay.
- 5 eBay serves the user the consent form for your application.
- 6 The user consents.
- 7 If your application hosts an AcceptURL, eBay serves the user back to that URL.
- 8 Your application requests a token for the user, who is identified in the request by the session ID.
- 9 eBay returns a token to your application.

Getting a token is a slightly different process for web-enabled applications (which can host AcceptURL and RejectURL) than it is for client desktop applications (which do not host URLs).

The following sections describe the processes for client desktop applications and for web applications.

- [“Option 1: Client/Desktop Applications” on page 122](#)
- [“Option 2: Web/Server Applications” on page 123](#)
- [“Getting a Token via FetchToken” on page 125](#)

Option 1: Client/Desktop Applications

Use this method of getting tokens if there is no web component to your application, or you do not wish to host an AcceptURL or RejectURL page.

In this process, some of the steps are completed by the user, some by eBay, and some by your application.

- 1 A user indicates an intention to use your application (for example, by clicking a Subscribe button in your application interface).
- 2 Your application makes a [GetSessionID](#) request to eBay, and receives a session ID ([SessionID](#)).
[GetSessionID](#) is described in reference detail at “[GetSessionID](#)” on page 866.
- 3 Your application uses the session ID to construct an eBay sign-in URL that sends the user to the eBay sign-in page and consent form.

Use the following URL to redirect a user to the eBay sign-in page:

`https://signin.ebay.com/ws/eBayISAPI.dll?SignIn&RuName=RuName&SessID=SessionID`

The URL must include `&SessID=SessionID` and `&RuName=RuName`. `SessID` identifies the user and `RuName` the application. The `SessionID` must be URL-encoded in the sign-in URL.

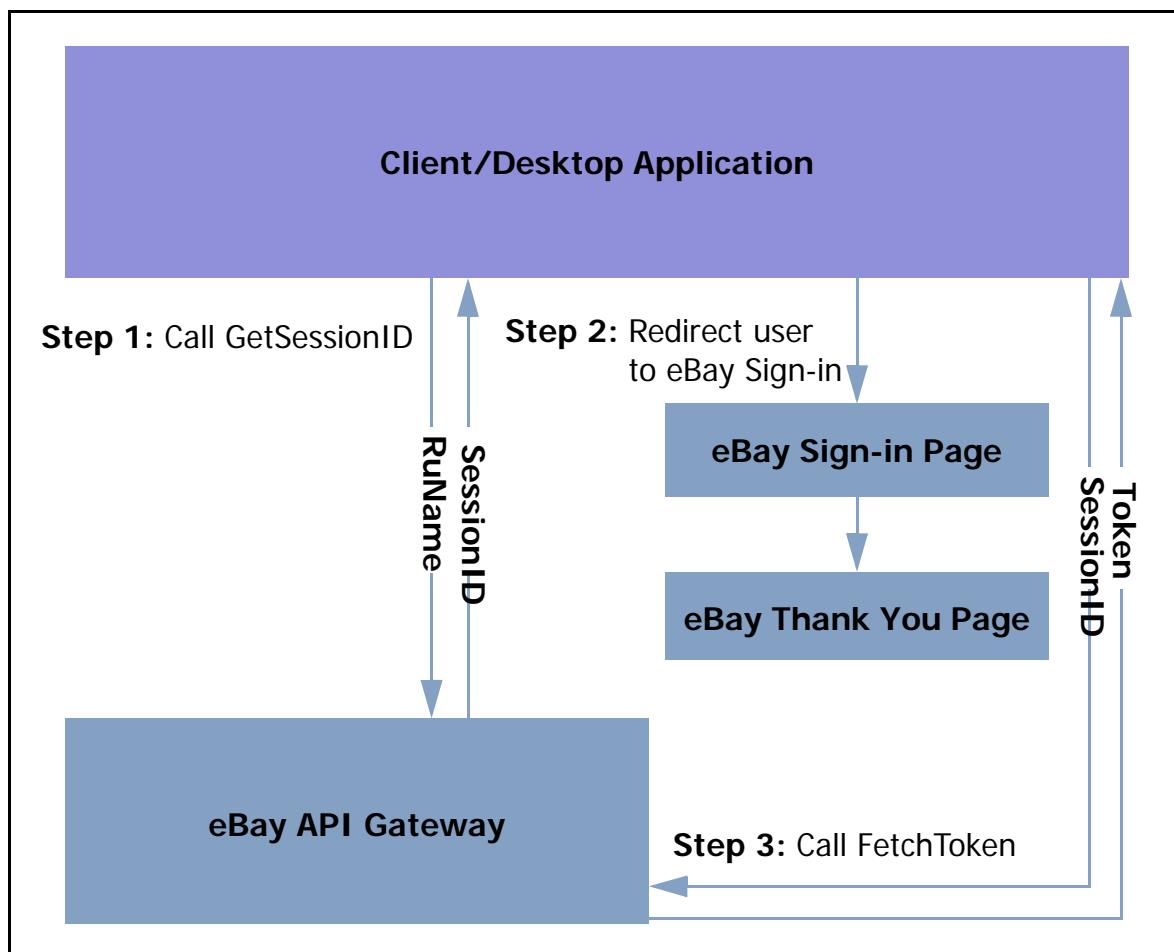
The equivalent URL for the Sandbox is:

`https://signin.sandbox.ebay.com/ws/eBayISAPI.dll?SignIn&RuName=RuName&SessID=SessionID`

- 4 The user signs in on the eBay sign-in page.
- 5 eBay forwards the user to your application’s consent form.
- 6 The user consents or rejects your application.
- 7 eBay forwards the user to an eBay page that confirms their action (they are thanked and told, “You may close this window”).
- 8 Your application calls [FetchToken](#) to retrieve the user token

See “[Getting a Token via FetchToken](#)” on page 125.

Figure 6-1 Option 1: User Sign-in Flow Ends on eBay; Client/Desktop Application Uses FetchToken to Retrieve Token



Option 2: Web/Server Applications

Use this method for getting tokens if your application has a web component and can respond to being redirected to the URLs specified on your [MyAccount](#) page.

In this process, some of the steps are completed by the user, some by eBay, and some by your application.

- 1 A user indicates an intention to use your application (for example, by clicking a Subscribe button in your application interface).
- 2 Your application makes a [GetSessionID](#) request to eBay.
[GetSessionID](#) is described in reference detail at “[GetSessionID](#)” on page 866.
- 3 Your application receives session ID ([SessionID](#)).

- 4 Your application constructs an eBay sign-in URL that sends the user to the eBay sign-in page and consent form.

Sign-in URL for production tokens:

`https://signin.ebay.com/ws/eBayISAPI.dll?SignIn&RuName=RuName&SessID=SessionID`

Sign-in URL for Sandbox:

`https://signin.sandbox.ebay.com/ws/eBayISAPI.dll?SignIn&RuName=RuName&SessID=SessionID`

In the URL, `SessID` identifies the user and `RuName` the application. `SessionID` must be URL-encoded in the sign-in URL. `SessionID` is the session ID that was obtained by calling `GetSessionID`.

Tip: In a web application, you can construct an eBay sign-in URL using an HTML form that contains a Submit button. For example:

```
<INPUT TYPE=\"submit\" NAME=AUTHORIZE VALUE=\"Launch Auth & Auth\" "
      .   "onclick=\"window.open('https://signin.ebay.com/ws/
eBayISAPI.dll?SignIn&RuName=$RuName&SessID=$SessionID');\">\n"
```

Tip: Your application can use an additional parameter in the sign-in URL, `ruparams`, to hold user data that you want passed back to your application after the user consents. After user consent, eBay adds the `ruparams` information to the URL that directs the user back to your AcceptURL or RejectURL page. This data is not used in any way by eBay.

The data you pass in `ruparams` may consist of zero or more pieces of information, each in a variable name and URL-encoded value format. When eBay redirects to your application's AcceptURL or RejectURL, the value of `ruparams` is URL decoded into name-value pairs. For example, if the application sends this:

`...&ruparams=VarA%3DB12309%26VarB%3DABC123`

eBay returns this in the URL:

`...&VarA=B12309&VarB=ABC123`

- 5 The user signs in on the eBay sign-in page.
- 6 eBay forwards the user to your application's consent form.
- 7 In the consent form, the user consents or rejects your application.
- 8 eBay redirects consenting users back to the application's AcceptURL, and rejecting users to the application's Reject URL.

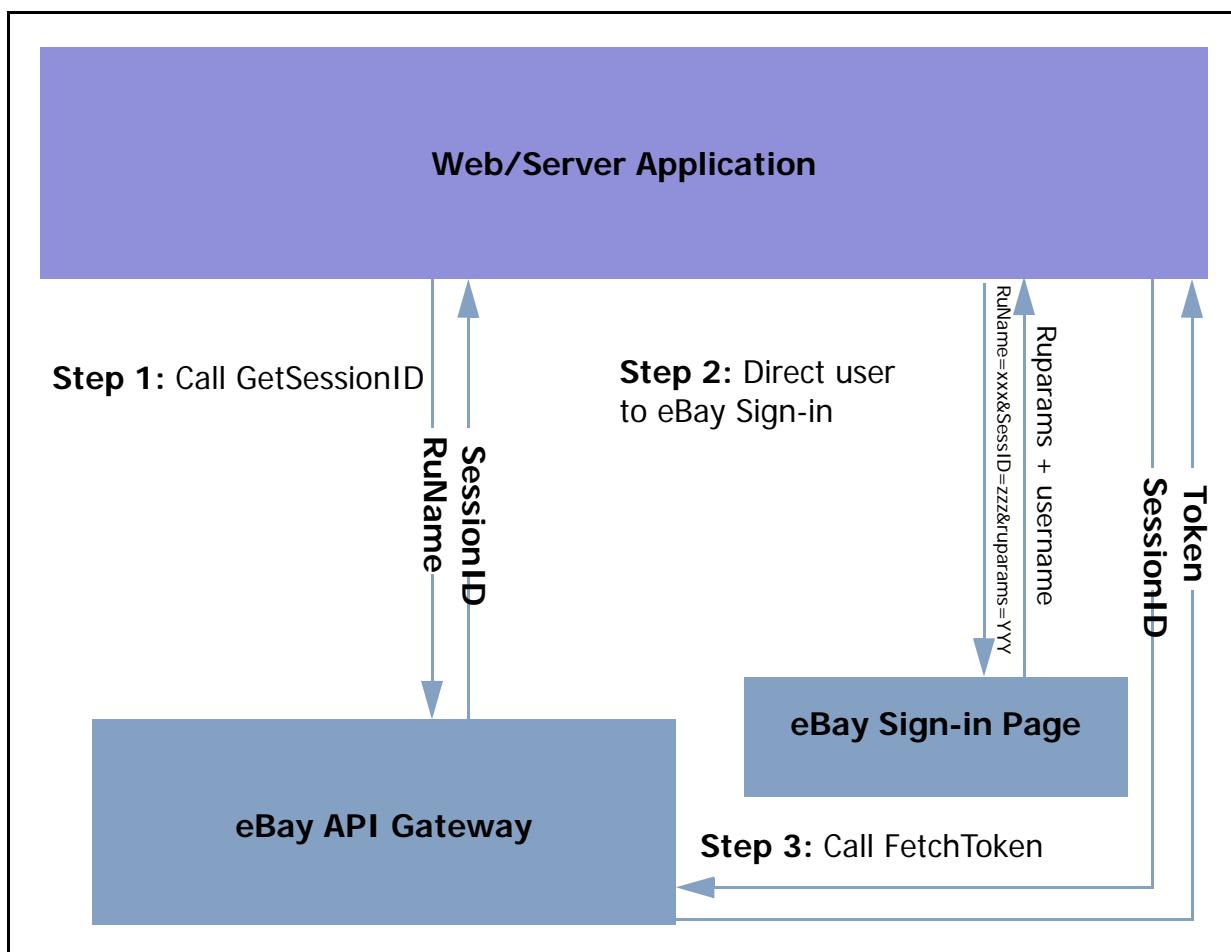
If you have passed user data in `ruparams` in the sign-in URL, eBay appends this data to the AcceptURL or RejectURL.

Your application should present rejecting users with a meaningful error or message indicating that the application will be unable to make calls to API calls in the user's name (that is, the user cannot use your application).

- 9 The application then calls `FetchToken` to retrieve the user token.

See “[Getting a Token via FetchToken](#)” on page 125.

Figure 6-2 Option 2: Sign-in Flow Returns User to Web Application; Application Uses FetchToken to Retrieve Token



Preparing the URL for User Sign-in

This information has been updated and moved. Please see “[Option 1: Client/Desktop Applications](#)” on page 122 for desktop applications, or “[Option 2: Web/Server Applications](#)” on page 123 for web applications.

The Form of the Accept Redirect URL

This information has been updated and moved. Please see “[Option 2: Web/Server Applications](#)” on page 123.

Getting a Token via FetchToken

Once the processes described in “[Option 1: Client/Desktop Applications](#)” on page 122 or “[Option 2: Web/Server Applications](#)” on page 123 have been completed, your desktop or web application use **FetchToken** to retrieve tokens for its users.

When a user hits the eBay sign-in URL your application constructed during the consent process, eBay generates a user token and stores it with a user's session ID for 48 hours.

If this is your application's first time using this process to obtain a token for a user, your application should wait 5-10 seconds before calling [FetchToken](#).

This section contains examples of using [FetchToken](#).

[FetchToken](#) is described in reference detail in “[FetchToken](#)” on page 798.

The SOAP API example below shows a [FetchToken](#) call with the requester credentials in the SOAP header:

Example 6-1 Fetching a Token Programmatically

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
    <soap:Header>
        <ebl:RequesterCredentials soapenv:mustUnderstand="0">
            <Credentials>
                <AppId>MyAppId</AppId>
                <DevId>MyDevId</DevId>
                <AuthCert>MyCertId</AuthCert>
            </Credentials>
        </ebl:RequesterCredentials>
    </soap:Header>
    <FetchTokenRequest xmlns="urn:ebay:apis:eBLBaseComponents">
        <SessionID>MySessionID</SessionID>
        <Version>613</Version>
    </FetchTokenRequest>
</soap:Envelope>
```

The XML API example below shows how to use the [FetchToken](#) call:

Example 6-2 Fetching a Token Programmatically

```
<?xml version="1.0" encoding="utf-8"?>
<FetchTokenRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <Version>613</Version>
    <RequesterCredentials>
        <DevId>MyDevID</DevId>
        <AppId>MyAppID</AppId>
        <AuthCert>MyAuthCert</AuthCert>
    </RequesterCredentials>
    <SessionID>MySessionID</SessionID>
</FetchTokenRequest>
```

For an explanation of how to use tokens, see “[Using Tokens](#)” on page 127.

Using Tokens

To authenticate the requester for a SOAP API call using a user token, pass the token in the header for the request. The SOAP API example below shows the header element that contains the token:

Example 6-3 User Token Header

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" 
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ebc="urn:ebay:apis:eBLBaseComponents">
    <soap:Header>
        <ebc:RequesterCredentials soapenv:mustUnderstand="0">
            ...
            <eBayAuthToken>
                ... USER TOKEN GOES HERE ...
            </eBayAuthToken>
        </ebc:RequesterCredentials>
        <Version>573</Version>
    </soap:Header>
    ... Call body ...
</soap:Envelope>
```

To authenticate the requester for an XML API call using the user token, pass the token in the `eBayAuthToken` element in the request XML. The XML API example below shows the full XML for the `GeteBayOfficialTime` call:

Example 6-4 User Token in a Call

```
<?xml version="1.0" encoding="utf-8"?>
<GeteBayOfficialTimeRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <RequesterCredentials>
        <eBayAuthToken> Token goes here </eBayAuthToken>
    </RequesterCredentials>
</GeteBayOfficialTimeRequest>
```

About Tokens

In this section:

- “[What is a Token?](#)” on page 127
- “[Life Span and Uniqueness](#)” on page 128
- “[Watching for Token Expiration](#)” on page 129

What is a Token?

The user token represents an authorization for your application to access eBay data (via eBay APIs) on behalf of an eBay user. A user grants an application a user token by giving consent via the eBay sign-in and consent form. The user does not provide his or her eBay username or password to the

application during this consent process. The user token accompanies API calls that the application makes on the user's behalf, authenticating the user and authorizing the application to make the API calls for the user. Before processing an API call made by a third-party application on behalf of a user, the API checks that the token is valid (not expired or revoked), and validates the application credentials in the header of the API request. A unique token is needed for every eBay user/application combination.

Life Span and Uniqueness

Typically, an application obtains a user token for each user via the user sign-in and consent process and stores the token for subsequent use. Tokens are valid for 18 months (unless intentionally expired early by the user on the user's My eBay page, by eBay upon suspicion of security compromise to the application or to eBay, or by the application—see “[Token Expiration Causes](#)” on page 131).

A user token is up to 2KB in length and is base 64 encoded for storing in a database table. It can contain these characters:

- a to z, A to Z, 0 to 9
- Asterisk, Slash, Plus (* / +)

For a single application, a user token provides user authentication for as many API calls as the application or user might make, and is functional across multiple sessions of the application. Applications are notified at least seven days before the expiration of a token at the end of its lifetime, so that they can initiate the process of generating a new token for the user.

A new token may be generated before the old one actually expires. The application need not wait until the exact expiration date specified in the notification. Typically, applications initiate the generation of a new token at some point between receipt of the expiration notification and the actual expiration date. The creation of a new token for a given application/user combination invalidates the prior token.

If a token expires and no new token is generated, the application will be unable to make API calls in that user's name using the expired user token, and cannot do so again until a new token has been generated. This failure is limited to that one combination of application and end-user. The same user could still make API calls using a different application—assuming there is a valid non-expired token for that user and that application. The user can still log in and operate on the eBay Web site normally. Only the user token expires, not the user's eBay account. Once a new token has been generated, the application will again be able to make function calls on that user's behalf with user authentication provided by the new token.

Should a user change his or her user name or password, an already-existing token for that user continues to function normally. The application will still be able to use the token for authentication for that user even after the user name or password change.

Watching for Token Expiration

To ensure that user tokens are secure and to help avoid a user's token from being compromised, tokens have a limited life span. A token is only valid for 18 months from the time of creation.

When a token expires, your application must redirect the user associated with the expired token to the eBay site to sign in again. After the user successfully signs in, a new token is generated and passed back to your application.

Seven (7) days before a token is due to expire, eBay returns the expiration date in the **HardExpirationWarning** field in the response of all calls the application makes on behalf of that user. When your application detects this field, it must redirect the user to the sign-in page on the eBay site by the date in this warning, or the token ceases to work as a means to authenticate that user.

eBay recommends messaging end users several months before the token expiration date to ensure the user has plenty of time to reauthorize your application. Keep track of the date on which you created the token, and assume that it should be valid for 18 months unless otherwise indicated by eBay.

Once a token has expired, all calls using that token will fail with error 932: “Auth token is hard expired.” To avoid unexpected user token failures, your application should include logic to handle this error.

Watching for Token Expiration

In SOAP-formatted calls, the seven-day warning is returned in the **HardExpirationWarning** element. The element is returned in the header of the response to an API call made within the 7-day warning period:

Example 6-5 Seven-Day Warning Header

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
        <soap:Header>
            <ebl:RequesterCredentials soapenv:mustUnderstand="0">
                <HardExpirationWarning>
                    ... DATE OF EXPIRATION HERE ...
                </HardExpirationWarning>
            </ebl:RequesterCredentials>
        </soap:Header>
        ... Call body ...
    </soap:Envelope>
```

The date in the **HardExpirationWarning** element is sent by eBay in GMT. However, if you are using Axis, it automatically converts the time to local time based on the system settings of the local machine. If the application uses another SOAP client, consult its documentation to determine whether this automatic conversion is done.

In XML-formatted calls, the seven-day warning is contained in an element named **HardExpirationWarning**. This element is returned in the body of the response XML for API calls

made within the 7-day warning period. The date and time of the impending expiration are contained in this element:

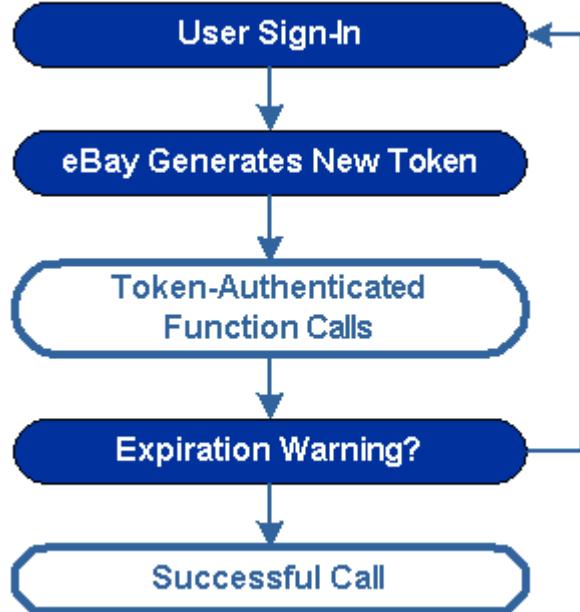
Example 6-6 Seven-Day Warning Response Element

```
<?xml version="1.0" encoding="UTF-8"?>
<GeteBayOfficialTimeResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2005-01-12T18:29:48.312Z</Timestamp>
    <Ack>Success</Ack>
    <CorrelationID>00000000-00000000-00000000-00000000-00000000-00000000-000000000000
    </CorrelationID>
    <Version>393</Version>
    <Build>20050110220901</Build>
    <HardExpirationWarning>2005-01-14 03:34:00</HardExpirationWarning>
</GeteBayOfficialTimeResponse>
```

eBay sets the amount of time between one expiration and the next. Currently the interval for expirations is 18 months from creation, but be aware that for security reasons this interval is subject to change at any time.

Rather than attempting to anticipate the next expiration (and neglect to watch for expiration alerts), an application should have logic that checks for these alerts with every API call it makes.

Figure 6-3 Token Expiration Flow



To successfully implement token-based authentication, an application must programmatically watch for, detect, and react to expirations as they occur.

Token Expiration Causes

The following are instances that cause token expirations to occur:

- **Token's expiration date has been reached**

Expiration warnings are returned starting seven days prior to the token's expiration date. The token is invalid after the expiration date. Any API calls made with the expired token fail with an error (error code 932).

- **User revokes the token**

Users are able to revoke tokens using the My eBay page on the website (My eBay > My Account > Site Preferences > General Preferences > Third-party Authorizations). When a user revokes a token for an application, that token expires immediately. Any API calls made with the expired token fail with an error (error code 16110).

- **eBay revokes a token due to security concerns**

If eBay detects suspicious activity that indicates a possible security compromise of an account or application, all affected user tokens are expired immediately. Any API calls made with the expired token fail with an error (error code 17470).



Part III

eBay Features Overview

eBay Concepts

133

Chapter 7

eBay Concepts

eBay is a marketplace in which one person (the *seller*) can securely sell something to another person (the *bidder* or *buyer*). That “something” is referred to as an *item* or *listing*. When a seller places an item on the eBay site, the act of placing the item is referred to as *listing the item*, and the resulting entry on the eBay site is referred to as a listing. A listing can contain one or more items, depending on the format of the listing (described later in this document).

On the eBay Web site, a seller who wishes to create a listing enters data about their item(s) by using a series of forms. These forms are collectively called *Sell Your Item* (due to a title that appears at the top of each form page on the US eBay site). The workflow is also sometimes referred to as the *Sell Your Item flow*.

Once the seller completes the Sell Your Item flow and submits the data to eBay, the listing can then be viewed by other users, with the hope that one of them will want the item (or items) and buy it (or them). In some listing types, users (known as *bidders*) lodge bids for an item (or multiple items, if the listing has more than one), competing against other bidders to have the highest bid and purchase the item, at which point bidder becomes *buyer*. In other types of listings, the user may purchase immediately, without having to compete with other bidders, and such a user is referred to as a buyer.

After initially creating a listing, a seller may need to modify (revise) a detail of the listing. To do this on the eBay Web site, the seller uses the *Revise Your Item* flow.

eBay Web Services enable developers to create applications that perform many of the same operations as the eBay site. API calls are used at particular points during the life cycle of a listing to serve particular purposes or the initiate particular actions.

The calls available on the eBay platform correspond to particular individual activities a user might engage in on the eBay site. Each call has a distinct purpose and may only be applicable at a particular point in the life cycle of an listing, such as when the item is first listed, when a buyer is considering the purchase of a listed item, or when the seller and buyer are completing the sale of an item.

The information in this document is intended to help a developer become familiar with the core concepts and terminology related to selling items and searching for items on eBay. It is organized into these sections:

- “[Basic Building Blocks](#)” on page 134—The API functionality is built around a set of fundamental concepts that the developer should know before creating robust and feature-rich applications using eBay Web Services.
- “[eBay User Roles](#)” on page 147—The users who participate in activities on eBay, whether through the site or an application, do so in a variety of roles. Each has a place in the overall workings of eBay.
- “[eBay Sites and Environments](#)” on page 149—eBay is not one site, but rather an integrated collection of sites that work together. There are some nuances and differences that a developer must consider.
- “[Affiliate Tracking Concepts](#)” on page 155—If you use affiliate-related fields, it is possible to obtain affiliate commissions based on calls made by your application.

Basic Building Blocks

The functionality of the eBay site and of the API is constructed around a number of basic building blocks. Each of these basic building blocks are represented by an object with a name that suggests the basic building block to which it corresponds. The two most important and commonly used basic building blocks are *users* and *items*. Users are the eBay members who sell and buy things on the eBay site, and items are the things they sell and buy. The other basic building blocks, also important, support the interactions between users and other users and between users and items.

Summary of Building Blocks

Item

Something sellable or marketable on eBay. See “[Items and Listings](#)” on page 136 for an introduction to items. See “[Listing Items](#)” on page 174 for information on how to sell items using the API.

Listing (noun/verb)

(noun) An entry on eBay that offers or advertises one or more items for sale.

eBay members (and these docs) sometimes use the terms *listing* and *item* interchangeably. Technically, a *listing* is the advertisement of what you’re selling, whereas the *items* are the objects that are being advertised within the listing. A listing can contain one or more items.

(verb) The action of creating a listing.

Listings can be purchased through competitive bidding (as auctions) or without bidding (fixed price).

See “[Items and Listings](#)” on page 136 for an introduction to listings. See “[Listing Items](#)” on page 174 for information on how to create listings using the API.

Category

A named location on eBay where items of a similar nature can be listed, such as “Fashion” or “Electronics”.

From the perspective of a user, categories are organized into a hierarchical structure (conceptually similar to the directories in a file system). See “[Categories \(Introduction\)](#)” on page 138 for the basic building blocks related to categories. See “[Categories](#)” on page 158 for details on working with categories in the API.

User

In the context of eBay Web Services, someone who has registered as a member of eBay.

Anyone can browse the eBay site without registering as a user; however, eBay Web Services functionality only supports registered users.

There are user roles such as bidder, buyer, seller, store owner and application developer.

See “[Users](#)” on page 145.

Order Line Item (Transaction)

The record of a buyer’s commitment to purchase one or more items from the same listing. A line item within an order contains information about one buyer’s purchase from one listing. (In the API, this is also sometimes called a *transaction*, for historical reasons.)

An order line item is created when:

- A buyer wins a competitive-bidding auction
- A buyer uses Buy It Now to purchase an auction item at full price before any bids are placed
- A buyer purchases one or more items from a fixed-price listing

If a seller offers more than two items for sale within a fixed-price listing, and multiple different buyers purchase those items, that listing will spawn multiple order line items (one from each buyer).

If a buyer purchases items from different listings by the same seller before paying for them, the seller can group the purchases into a the same *order* (which can form the basis for an invoice), where each purchase is reflected by an order line item within that order.

See “[Order Line Items](#)” on page 145.

Feedback

An eBay mechanism by which one user rates and provides feedback for the user on the other side of the order. Feedback and the cumulative Feedback Score helps an eBay user to determine whether or not they want to do business with that buyer or seller. See “[Feedback](#)” on page 145.

Feature

An option, capability, or workflow that enhances the user’s experience on eBay.

Users can take advantage of a number of useful features, such as shipping calculators, predefined payment options, advanced search capabilities, custom preferences, and other features offered on eBay.

Items and Listings

In the API, both items and listings are represented by the [Item](#) object. The same defining values a seller would supply when listing the item through an eBay Web site are supplied in the [Item](#) object. Thus, instead of entering the data through an eBay Web site, the seller can enter it through an application that uses the API. Programmatically supplying item data also lends itself to the creation of applications that have no user interface at all, such as an application that automatically lists items to the eBay site from an inventory database, all without direct human intervention.

The data a seller (or an application) supplies to define an item and its listing includes such things as:

- The title under which the item appears on the eBay site
- The price at which the seller wishes to sell the item
- The length of time (or duration) the item's listing appears on the eBay site
- The category in which the item can be found by buyers
- Specifications from which the costs of shipping can be calculated
- The methods by which the seller enables the buyer to pay for the item
- Where (geographical regions) the seller is willing to ship the item

The [Item](#) object is reusable. When used with operations such as [AddItem](#), the data in the [Item](#) object is passed to eBay where it is used to list or re-list the item. When used with such operations as [GetSellerList](#) and [GetItem](#), the [Item](#) object is populated with the listing data retrieved from eBay. Note, though, that not all elements of the [Item](#) object are used for all API call purposes. For instance, the [Item](#) objects returned by a [GetSellerEvents](#) operation each only have a subset of the data a [GetItem](#) operation would fill an [Item](#) object with. So, while the [Item](#) object is reused in a number of places in the API, it is not always reused in exactly the same way.

For technical how-to information related to items (listing, retrieving, managing) see the Part IV: Managing Listed Items chapters:

- “[Listing Items](#)” on page 174
- “[Retrieving Items](#)” on page 502
- “[Managing Item Listings](#)” on page 375

Listing Types (Formats)

eBay offers several listing types to accommodate the selling needs, styles, and approaches of different sellers. The table below describes the listing types available.

Not all listing types can be created with [AddItem](#) and related calls, and some listing types may be available on some sites but not in the API. Some listing types have prerequisites the seller must meet before using them (such as a minimum aggregate feedback score). For information about prerequisites for using each listing type, see the online help on the eBay site.

The listing types differ in how a sale is made (progressive bidding or immediate purchase). Other differences include the number of items that can be offered for sale in a listing, the number of days a listing may be active, and how and when the listing ends.

Table 7-1 Item Listing Types

Listing Type	Definition
Chinese auction	<p>An auction listing in which only a single item (or lot) can be offered for sale.</p> <p>The listing is active for the period of time the seller selects when the item is listed and ends when that period has passed (or if a buyer uses the Buy It Now feature if offered for the listing). Bidders attempt to purchase the item by lodging progressive bids, competing with other bidders to have the highest bid.</p>
Real Estate Ad	<p>When the duration of the listing has passed, the listing ends and a winning bidder is determined. This is the bidder with the highest bid when the listing ends. That user is then able to purchase the item at that bid price from the seller. If the seller is an eBay Store owner, the item can be listed in both a general eBay category and a custom category in the seller's Store. An auction listing may be active for (i.e., may have a duration of) 1, 3, 5, 7, or 10 days (or 30 days for a real estate listing).</p>
	<p>Buy It Now option: When the seller first creates an auction listing, he or she may implement the listing with a Buy It Now option. When this option is available, the first user who wishes to purchase the item may do so at a fixed-price that the seller specified when the item was listed. If the user exercises this option, the purchase occurs immediately and the listing ends. Alternately, the user may simply enter a normal bid for the item. In this case, the Buy It Now option becomes unavailable for the listing and there follows the usual process of progressive competitive bids and any bidder may wind up as the winning bidder. If the listing has a reserve price, the Buy It Now option does not disappear at the first bid, but remains until the reserve price is met (and a second, third, or even subsequent bid might be required before one of them meets or exceeds the reserve price).</p>
	<p>On some sites, the Buy It Now price for an auction may remain available for certain categories even after a bid is placed. See the "Longer Lasting Buy It Now" community board post below for more details and the latest list of eBay categories that support this behavior. The eBay Spain site supports a longer-lasting Buy It Now option for all categories.</p>
	<p>Longer-Lasting Buy It Now</p>
	<p>http://forums.ebay.com/db1/thread.jspa?threadID=2000449591</p>
	<p>Non-purchase advertisement listing used to generate leads. Instead of bidding, interested parties submit contact information for learning more about the listed item. No actual purchase is made through the listing.</p>
	<p>A Real Estate Ad listing may have a duration of 30 or 90 days. On some eBay sites, all Real Estate listings require you to include standardized descriptive fields (called <i>Item Specifics</i>) in the listing. For more on real estate listings, see "Working with eBay Real Estate Listings" on page 557.</p>

Table 7-1 Item Listing Types (Continued)

Listing Type	Definition
Basic fixed-price listing	<p>A fixed-price listing is a listing in which there is no progressive bidding. Buyers purchase one or more items from the listing at a static price. These listings may offer one or multiple items for sale.</p> <p>A Basic Fixed-Price listing containing multiple items does not end when a buyer purchases one of the items. Instead, the listing ends when the last item is sold or when the listing's duration has passed, whichever occurs first. The listing may end with some or all of the items unsold.</p> <p>The available durations for the fixed-price items are 3, 5, 7, 10, and 30 days, or a seller may create a "Good Til Cancelled" (GTC) listing. GTC listings have a duration of 30 days, but the listing is automatically relisted for another 30 days if it ends with at least one item unsold. This automatic relisting continues indefinitely until the seller cancels the listing or the last item is sold.</p> <p>A fixed-price listing can be of the non-variation variety (multiple quantities of an identical product) or it can be multi-variation (multiple quantities of a similar item with slightly different specifications). A multi-variation listing is often used with clothing, so sellers can offer different sizes and colors. Regardless of whether a fixed-price listing is non-variation or multi-variation, as its name indicates, all items purchased from a fixed-price listing will have the same item price.</p>
Personal Offer	<p>Special type of listing extended by the seller to one specific buyer.</p> <p>This listing type is specific to the second chance offer feature (see "Making Second Chance Offers for Items" on page 426). It is available on the eBay site and via API with <code>AddSecondChanceItem</code>. The listing is only available to and accessible by the user to whom the seller extended the offer (and the seller). Only this offer recipient may purchase the item. As only one person may buy the item, there is no bidding and the purchase acts much the same way as for a fixed-price item. A Personal Offer listing may have a duration of 1, 3, 5, or 7 days.</p>

Categories (Introduction)

On the eBay site, *categories* are a hierarchical set of groups in which items of a similar nature are listed. Listing an item in the proper category greatly increases the chances that an interested user will find and buy the item. If a camera is listed for sale in a category intended for baby clothes, the chances are small someone interested in purchasing a camera would see it. Buyers benefit from categories in that they can go to a specific category to find something they wish to buy (or to simply browse the available items and "impulse shop"), because they know that the category contains only the type of items they are interested in and the category is the designated place to find them.

Categories vary from one eBay site to the next. That is, the US eBay site has different categories than, say, the Germany eBay site. Many of the categories serve the same purpose on all sites (though they are named in the language of the site), but a site also contains categories that are only of regional interest or of interest to a particular culture.

Each category has a unique identifier (for such purposes as listing an item). But these category IDs are only unique to the global eBay site on which they are hosted. For example, the ID for the category with sports memorabilia is different on the US site than the ID for the corresponding category on the France eBay site.

An item can be listed in at most two categories (called the primary and secondary categories). For example, a book on tape might be listed in Audiobooks as the primary category and Cassettes as the secondary category. In most cases, the seller will choose as the primary category the category in

which they think buyers would most likely browse or search for the listing. On some sites, certain categories are required to always be the primary category if they are used. For example, when listing a ticket for sale on the US site, the Tickets category must be selected as the primary category.

For technical how-to information related to categories (retrieving and storing category data) see “[Categories](#)” on page 158. Also see the chapter on [GetCategories](#), “[GetCategories](#)” on page 805.

Retrieving Categories

In the API, the hierarchy of categories for a site is retrieved from eBay using [GetCategories](#). An application can retrieve the entire category hierarchy or only that part of it that is child to a particular parent category. The category hierarchy is versioned so that an application knows when the category hierarchy has changed and downloads the hierarchy only when it is needed, saving unnecessary download time, activity, and bandwidth.

The category hierarchy for a site is embodied in the [CategoryArray](#) object. A single category is represented by a [Category](#) object. A [CategoryArray](#) object is simply a collection of [Category](#) objects.

After retrieving category data, an application will store this data for later use listing or retrieving items based on specific category IDs. While an application may use a transient storage method where the data only exists for the duration of the application’s session, a persistent method of storing category data is preferred. If transient storage is used, the application must retrieve the category data from eBay each time the application is run. This is very inefficient and can be very time-consuming.

Using Categories

Categories (or, more specifically, the IDs of categories) are used in listing items. When a seller lists a new item, the seller must specify which category the item should appear in. Similarly, when re-listing an item the seller may specify a category other than the one it was originally listed in.

An application can also retrieve a list of items various sellers offer within a specific category. This functionality is analogous to the browse screens of listings for categories seen on the eBay site.

Choosing a Listing Option (Standard Way or with Product Details)

In addition to defining the basic properties of an item, some categories support options that let you fill in item descriptions more quickly by using stock product information from an eBay catalog.

The options that are available depend on the item’s category:

- Listing the standard way

With this option, you enter all the descriptive information about the item yourself. This option is available in all categories.

- Listing with catalog product details

With this option, you identify a stock catalog product that is similar to your item, and eBay uses that product’s stock content to fill in a portion of your listing’s details and to add other information to supplement your listing description. The catalog contains descriptive information only (no actual physical products). This option is available in certain categories (which you can determine based on an API call, as described later in this document).

Listing with Product Details

To include product details in a listing, you pass in an eBay Product ID (EPID), which is a unique reference number that eBay defines for all items in our catalogs.

If you don't already know the EPID (this is common the first time you list a particular type of item), you can use an industry-standard identifier, such as:

- A Universal Product Code (UPC)
- The product's brand and manufacturer part number
- An ISBN for items like books, DVDs, music CDs, and video games
- An EIN (similar to an ISBN, in Europe)
- For tickets, the event name, venue, and date.

eBay then matches that identifier to a product in the catalog to determine the EPID for you, and adds the product details to your listing.

Listing with Item Specifics

As the seller fills in their listing description, they can increase the likelihood of buyers finding the item by including common search terms in the description. In many categories, there are well-known, category-specific aspects of items that buyers want to search on. For example, in book categories, buyers typically want to search by subject or author. In event ticket categories, buyers may want to search by particular event types (e.g., concerts).

To make it easier for sellers to determine these category-specific search keywords (like "Subject") and include them in their listings, eBay provides a special section in the item description called *Item Specifics*. Item Specifics are well-known, predefined characteristics that are common to all items in a particular category. For example, if you are listing in the Fiction & Nonfiction Books category, "Subject" and "Publication Year" are some Item Specifics for a book.

When you list the standard way, many categories provide Item Specifics that you can fill in. In most cases, filling them in is optional. In a few categories, Item Specifics are required.

When you list with product details, Item Specifics are always included in the listing.

[Figure 7-1](#) shows an example of Item Specifics that a seller has filled in while listing an item on an eBay Web site.

Figure 7-1 Listing the Standard Way with Item Specifics

Item specifics : Fiction & Nonfiction Books

Subject
Fiction & Literature

Fantasy

Condition
New

Format
Softcover

Publication Year
1999

If you do not know the Publication Year, please leave blank.

If you are new to the API and you are developing and testing your first listing application, you might start by implementing support for listing the standard way with a basic description. Then, after you have created a few different kinds of listings successfully, add support for listing with Item Specifics and product details.

Typically, end-user applications that support Item Specifics need to implement one or more of these capabilities:

- When a seller is listing an item in an attributes-enabled category, display the available characteristics for that category.
- When a user is searching for items, display the available characteristics for the selected category.
- When a seller is ready to add, revise, re-list, or verify the addition of an item, pass the selected attributes and values to eBay.
- When displaying a listed item to a user, display the attributes that have been set on the item.

About Item Specifics on the eBay Web Site

Most listings support certain standard properties that you can use regardless of the category. For example, you can choose to use a bold title for just about any kind of listing.

Within certain categories, there may be additional, category-specific aspects of items that buyers want to search on. For example, in book categories, buyers may want to search by format (e.g., softcover) or publication year. In event ticket categories, buyers may want to search for particular event types (e.g., concerts) or venues.

To make it easier for sellers to predict the category-specific search keywords (like "Softcover") that buyers will use and include them in their listings, eBay provides a special Item Specifics section in the item description. The Item Specifics section contains a set of well-known, predefined fields that are common to all items in a particular category. For example, if you are listing in the Fiction & Nonfiction category, Format and Publication Year are some Item Specifics for a book. If you are listing in the Event Tickets category, Event and Venue would be some Item Specifics for a ticket.

Many categories support Item Specifics, and this feature is being added to more categories on a regular basis.

Figure 7-2 Item Specifics in the eBay User Interface



By using Item Specifics, sellers worldwide can describe their items using standardized search keywords. Buyers and sellers can use Item Specifics to find items more easily when searching eBay. For example, in the browsing flow, buyers use product finders to filter listings according to Item Specifics. Item Specifics also help buyers ascertain the details of items more easily when viewing items. See the eBay online help for more information about how buyers and sellers use Item Specifics on eBay.

Item Specifics Online Help

http://pages.ebay.com/help/sell/item_specifics.html

In the Sell Your Item flow, the way you enter Item Specifics varies depending on the listing option you choose:

- When you list the standard way, many categories support Item Specifics, but filling them in is usually optional. A few categories (such as Real Estate in some countries) require you to fill in certain Item Specifics in order to list the item. Some categories do not support Item Specifics at all. In any case, you also need to provide your own free-form description of the item.

When you list the standard way with Item Specifics, you fill in the values of the Item Specifics yourself (i.e., eBay does not fill them in for you). For example, if you are listing a book, you fill in the Subject field by choosing from a list of subjects, and you fill in the Publication Year field by entering a value. In some cases, the value you select for one field will control the availability of other fields or the choice of values for certain fields. For example, for a Tickets listing, the state or province you choose controls the list of names that will appear in the City field.

For listing with item specifics, see “[Working with Custom Item Specifics](#)” on page 481.

- When you list with product details, Item Specifics are always available. The difference is that eBay fills in many of the Item Specifics for you based on stock product information that you choose from an eBay catalog.

To search with catalog product details, see [FindProducts](#) in the Shopping API.

For some products, some Item Specifics might not be pre-filled, so you can fill them in yourself or select them from a list. For example, if you are listing a book, you can fill in a Condition field by selecting “New” or “Used.”

Figure 7-3 Standard Item Specifics vs. Pre-filled Item Specifics

Item Specifics (Standard flow)		Item Specifics (Pre-filled Item Information flow)	
Item specifics : Fiction & Nonfiction Books		Item specifics : Fiction & Nonfiction Books	
Subject	Fiction & Literature	Title:	Harry Potter and the Chamber of Secrets
	Fantasy	ISBN:	0439554896
Condition	New	Publisher:	Arthur A Levine
Format	Sofcover	Format:	Hardcover
Publication Year	1999	Publication Year:	2003
If you do not know the Publication Year, please leave blank.			

About Listing with Product Details on the eBay Web Site

As described earlier in this chapter, items in certain categories can have item specifics whose values are well known and don't vary from listing to listing. For example, in the Movies category, buyers would expect a popular movie's title, rating, and film credits to be the same no matter who the seller is. Instead of filling in this standard information for every listing, a seller can create a listing in which this information has already been partially filled out (pre-filled). This data is called *product details* (formerly also called *Pre-filled Item Information*).

You can use key attributes such as ISBN, UPC, or Manufacturer Part Number (depending on the product type), as well as keywords like Title to pull up stock listing information on items in categories that support this feature. You can then select content, including stock photos, descriptions, and supplemental information (like song lists, film descriptions, etc.) that will be added to a listing automatically. At the time of this writing, Media categories (Books, DVDs & Movies, Music, and Video Games) and Consumer Electronics categories (Digital Cameras, Cell Phones, and PDAs) support product details.

The use of product details is optional. Thus, if a category supports product details (i.e., it is catalog-enabled), it only means that you have the option to list with product details. It does not mean you must use this feature to list in that category.

Buyers and sellers can use product information to list and find similar items more precisely and to ascertain the details of items more easily when viewing items. By using product details, sellers worldwide can describe their items more fully in fewer steps and in a way that makes their listings easier to find.

Product details do not replace a seller's own item description; it just adds more information to the listing. Sellers still assume full responsibility for the content of the listing and the item offered.

View Item page: Once the item has been listed, the product details is included on the View Item page along with the seller's usual title and description.

[Figure 7-4](#) shows a View Item page for a book listing that includes stock product information. In this case, the user used the stock photo instead of providing their own picture, and they specified that they wanted to include more stock information in their listing in addition to the pre-filled Item Specifics.

Figure 7-4 View Item Page for a Listing with Product Details

Seller's listing title

Stock photo

In this case, no PictureURL was specified

Pre-filled book title attribute

Stock photo

Harry Potter and the Chamber of Secrets by... Item number: 3500647723

You are signed in This item is being tracked in [My eBay](#)

Revise your item Want to sell more quickly and efficiently? Learn about how [Turbo Lister](#) can help you save time.

Sell a similar item Want to increase your chances of selling this item? Find out how you can attract more buyers with [optional features](#).

Starting bid: US \$1.00 Place Bid >

Time left: 6 days 2 hours 30-day listing Ends Oct-23-03 14:21:06 PDT

History: 0 bids

Location: San Jose, CA United States /San Jose

Seller information
qatuser1 (3614) Feedback rating: 3614 Positive feedback: 85.4% Registered Jul-27-01 in United Kingdom
[Read feedback reviews](#) [Ask seller a question](#) [View seller's other items](#) [Visit this seller's eBay Store!](#) [Purchase Protection](#)

Description (revised)
Harry Potter and the Chamber of Secrets Item Specifics - Fiction & Nonfiction Books
Author: J. K. Rowling Publication Year: 1999
Publisher: Scholastic Paperbacks Special Attributes: --
ISBN: 0439064864 Language: English
Format: Hardcover Subject: --

Stock Photo
 Part of this page Copyright 1995 - 2003 Muze Inc. All rights reserved.
[Additional information](#)

Great book for kids and adults alike!

Additional Information about Harry Potter and the Chamber of Secrets
Part of this page Copyright 1995 - 2003 Muze Inc. All rights reserved.

Synopsis
In the second book about the adventures of young Harry Potter, wizard-in-training, our hero returns to school after an especially lousy summer vacation with his family only to find himself--and his friends--in the middle of a mystery involving a 50-year old secret and a missing heir.

Details
Publication Date: 1999-06-01
Series: Harry Potter (Cloth)
Illustrator: Mary Grandpre

Size
Length: 341 pages
Height: 9 in.
Width: 6 in.
Thickness: 1 in.
Weight: 26 oz.

Publisher's Note
When the Chamber of Secrets is opened again at the Hogwarts School for Witchcraft and Wizardry, second-year student Harry Potter finds himself in danger from a dark power that has once more been unleashed on the school.

The seller, qatuser1, assumes full responsibility for the content of this listing and the item offered.

Item Specifics

Seller's description

Additional pre-filled information

(Same as View Details in Sell Your Item)

Stock Photo

Find MORE great stuff at my eBay
[Visit qatuser1 store today!](#)

Users

An eBay user has a unique identifier by which the user can be referred. This user ID is unique across all of the eBay sites, US and global.

In the API, a user is represented by the [User](#) object. This object contains general information about a user, such as the user ID, whether the user is a new eBay member, and the eBay site on which the user registered. It also contains information that plays a part in buying or selling an item, like the user's aggregate feedback score, an indication of whether the user is an eBay Store owner (and the Store's location), and an indication of whether the user is a Power Seller. The data for a user can be retrieved using [GetUser](#), which populates a [User](#) object with the data it retrieves from eBay.

For more information on retrieving information about a user from eBay, see “[Retrieving User Information](#)” on page 513.

See also “[eBay User Roles](#)” on page 147.

Order Line Items

The [Transaction](#) API object contains the information of one order line item. An order line item is created once a buyer commits to buy an item.

Auction listings only result in one order line item, and this occurs either at the end of the listing with a winning bidder, or when the buyer uses the Buy It Now option. Fixed-price listings can spawn multiple order line items from multiple buyers if the seller offers multiple items (variation or non-variation) in the same listing.

The checkout flow encompasses the period of time between creation of the order line item (transaction) and shipping of the item. During the checkout flow, interactions between the seller and the buyer affect data values for the corresponding order line item. For instance, the [CheckoutStatus](#) container in order and order line item (transaction) retrieval calls has fields that indicate the current status of the order line item (checkout is complete (paid for) or incomplete (not paid)). The base amount of the order line item is shown in the [Transaction.TransactionPrice](#) field and the shipping and handling costs are shown in the [Transaction.ActualShippingCost](#) and [Transaction.ActualHandlingCost](#) fields. If PayPal is the selected payment method for the item, then PayPal would set the [Transaction.TransactionPrice](#) field at the end of the checkout process.

For Trading API applications, an order line item is represented by the [Transaction](#) object. To retrieve order line item data for a particular listing, you use [GetItemTransactions](#). This call can result in the return of zero or multiple order line items (transactions). To retrieve all single and multiple (Combined Payment) order line items for a seller's account, you use [GetOrders](#).

For technical how-to information related to order retrieval calls, see “[Completing the Sale](#)” on page 415.

Feedback

eBay provides a system by which a user may rate another user when they interact with them, enabling still other users to know how well or how poorly the interaction went. This rating system is known as *feedback*. The feedback system is a quantitative expression of the desirability of dealing

with a particular eBay user as a buyer or a seller. A user's feedback score is essentially that user's reputation on the eBay site.

After the successful conclusion of an item's sale, the two users who participated in the sale each have the opportunity to leave feedback about the other participant. The buyer can leave feedback about the seller and the seller can leave feedback about the buyer. Some listing types permit the seller to offer multiple items, which may lead to multiple buyers purchasing the items. In those cases, one buyer-seller and seller-buyer feedback exchange can be made for each buyer (the seller can get feedback from multiple buyers and can leave feedback for multiple buyers).

Feedback consists of several types: positive, negative, neutral, and withdrawn. A given user's aggregate feedback score is a numeric value that is calculated by adding up all of the positive feedback that user received and subtracting from it the sum of all negative feedback (neutral feedback neither adds to nor subtracts from a user's aggregate feedback score). This score is then displayed for all other users to see (see exception below). A seller's feedback score appears on the pages for items listed by that seller. A buyer's feedback appears with bids that buyer places on items. And, the eBay site provides a page where any one registered user may view the feedback score and history for any other eBay user. In API and SDK applications, a user can leave feedback for another user or retrieve feedback of another user.

A user may specify that his or her feedback is private (a preference setting on that user's My eBay page). If the user does this, other users cannot see that user's feedback score or history. It is hidden in bids, and the feedback summary page for that user does not show the data. A seller whose feedback profile is private will receive an error when the seller tries to list an item. The seller can list items after making the feedback profile public.

In addition to the subjective feedback score one user can leave for another, the submitting user can also supply a textual comment clarifying or expanding on the raw score. For example, a buyer might accompany a positive feedback with a comment like "Great seller! Would do business with this seller again."

Feedback is entirely voluntary, and one order participant leaving feedback does not require that the other participant also leave feedback.

On the eBay site (but not through the API), members can mutually withdraw feedback for an order if both parties involved in the order agree to do so. In this case, feedback left by both parties is withdrawn at the same time. Withdrawn feedback does not affect the Feedback Score for either user, but the original comments left remain.

In applications, leaving feedback for another user is done with [LeaveFeedback](#) and retrieving feedback about another user is done with [GetFeedback](#).

For technical how-to information related to feedback (leaving and getting feedback) see "[Leaving and Retrieving Feedback](#)" on page 432.

eBay User Roles

No matter what role a user plays, the user must be a registered eBay user.

Example 7-1 Types of Users

```
Registered eBay User
  |
  +- Application Developer (registered also with eBay Developers Program)
  |
  +- Seller (currently selling at least 1 item)
    |
    +- Store Owner (also subscribed to eBay Stores System)
  |
  +- Bidder
  |
  +- Buyer
```

The role a given eBay user plays at any given juncture is not the only role that user can play at other times. A user may be a seller (one user role) in one order and still be a buyer (another user role) in another order. An eBay membership is generic in terms of user roles. That is, there is no membership type specific to sellers and another to buyers. A user simply acts in one role or capacity as they need or want to do, under the same user identity.

- ["Registered eBay Users" on page 147](#)
- ["Application Developers" on page 147](#)
- ["Sellers" on page 148](#)
- ["Store Owners" on page 148](#)
- ["Bidders and Buyers" on page 148](#)

Registered eBay Users

Anyone may visit an eBay site, browse items offered for sale, peruse the available jobs list, and read the online help for the site. But to do anything meaningful—like selling and buying items—a user must become a registered eBay user. Becoming an eBay member is free and registration is easy. Registering supplies eBay with a means to contact the user and results in the user having a unique user ID. This ID is used to log into an eBay site, after which a user (if otherwise qualified) may sell items, buy items, or develop applications.

Interactions on eBay are relatively anonymous. Other users only see that person's user ID, never their real name. When a seller lists an item, other users see that user ID on the View Item page. Similarly, when a user places a bid on an item, other users see the bidding user's ID, not the name. Interactions on eBay are not, however, completely anonymous. eBay does have a real name and contact method for each user. When conflicts arise or mischief is done, eBay helps settle the problem between the two users who only know each other by their respective user IDs or contacts the offending user directly.

Application Developers

An application developer is one who develops API applications or the representative of a company that develops such applications. In the case of the latter, the company may have hired one or more

developers to write the program's code, but the user referred to as "application developer" in this context is the one who interacts with the eBay Developers Program.

To develop applications that interact with eBay through the API or the SDK, a user must register with the eBay Developer's Program.

Sellers

A *seller* is a user who is acting in the role of someone selling an item on eBay. The seller of an item is the eBay user who defines and lists a new item and is the user the buyer pays when purchasing the item. A user who sells items is not limited to this role and may at any other time also buy items.

To sell an item on eBay, a person must be a registered eBay user.

Store Owners

A store owner (store subscriber) is an eBay user who has subscribed to the eBay Stores system and has an eBay Stores storefront (for more on eBay Stores, see "[eBay Stores](#)" on page 154).

To own an eBay Store and sell items in that Store, a person must be a registered eBay user.

Bidders and Buyers

Bidders and buyers are the other side of eBay auctions from sellers. These are the users who are either buying an item or who are attempting to do so by becoming the winning bidder for an item.

A *bidder* is a user who makes a bid on an item in an auction listing. Competing with other bidders in a series of progressively higher bids, a bidder vies to be the highest bidder when the listing ends. A bidder who is the high bidder when an auction listing ends is deemed the *winning bidder* and is able to purchase the item. The winning bidder makes the transition from bidder to buyer and can then purchase the item from the seller.

A *buyer* is a user who has or is able to purchase an item from the seller. In auction listings, a user is only able to purchase the item after the listing has ended and the user is one of the winning bidders for the item. But for fixed-price listings (and auction listings with an active Buy It Now option), a user simply purchases the item at the specified fixed price.

To bid on and buy items, a person must be a registered eBay user.

For more on competitive-bidding, fixed-price, and Buy It Now listings, see "[Listing Types \(Formats\)](#)" on page 136.

eBay Sites and Environments

eBay includes an integrated collection of sites. In addition to the US and other sites, eBay includes a Sandbox environment for your use.

["The Sandbox and Production Environments" on page 149](#)

["The US and International eBay Sites" on page 150](#)

["PayPal" on page 153](#)

["eBay Motors" on page 153](#)

["eBay Stores" on page 154](#)

["Shopping.com" on page 154](#)

["Half.com" on page 155](#)

The Sandbox and Production Environments

Trading API applications can make calls in the Sandbox environment and the Production environment. The difference in impact of your application's activities is critically important.

Sandbox Environment

The Sandbox environment is a test environment, in which developers can test applications. Items listed and sold in this environment are not real. The users who act as sellers and buyers are fictitious users that developers set up for participation in auction simulations to test applications. Similarly, the funds used in the selling and buying of items are not real. Listing an item incurs a fee for a fictional test user, but come from a reserve of fictional money assigned to the test user.

In other words, in the Sandbox, you can list items, end items, leave feedback, and so on, as part of testing your application, and none of it "counts." That is, you won't be charged for listing or buying the test items you use.

You can access the Sandbox programmatically or you can browse the Sandbox Web site by means of a browser. You can create one or more test users and set them up as sellers to simulate listing items to the Sandbox through your application or via the Sandbox Web site. You can create additional users to act as buyers to bid on those items via the Sandbox Web site.

Sandbox Environment

<http://sandbox.ebay.com>

Production Environment

Production is the live eBay site. Item selling and buying is done by real eBay users (through the online eBay site as well as applications). The money used in selling and buying items is real. And listing items incurs listing fees that the seller must pay.

An eBay-enabled application that is sending real data to the eBay Web site is executing calls in the Production environment. A user who views the eBay Web site through a browser is viewing data in the Production environment.

When you're sure your application is working the way you want it to, you can start making calls to Production. Everything "counts." That is, you will be charged listing fees, or for buying items.

Negative feedback will stay on your record. Testing on your application must be complete before you move it to Production.

Facts About the Sandbox and Production Environments

Here is additional information about the two environments, and facts an application developer must consider:

- An item listed in the Sandbox never appears in the Production environment. Nor do items listed in the Production environment ever appear in the Sandbox.
- A test user set up in the Sandbox only works within the Sandbox environment and cannot be used in or accessed from the Production environment. Similarly, a user in the Production environment cannot operate in or be seen from the Sandbox. See “[Creating a Test User](#)” on page 63 for more information about test users.
- Each environment has a dedicated URL that applications use for sending data to and retrieving data from eBay. This enables an application to be designed, programmed, and tested entirely within the Sandbox, and simply changing the target URL to that of the Production environment enables the application to work in the Production, as-is, with real items and real users. See “[Routing the Request \(Gateway URLs\)](#)” on page 87 for the current URLs.
- The key set for operating in the Sandbox is different from the key set for operating in Production. See “[Checklist for Going Live](#)” on page 67.

The US and International eBay Sites

eBay consists of a number of sites across the globe that all enable users to sell and buy items. The difference between these sites being that each is dedicated to a particular geographical region and uses language of the region for the eBay user interface and item listings.

Sellers list to a particular site to make their items available to the buyers who are registered on and log in to a particular geographical site. Sellers can list an item to more than one site to make the item available to a broader audience. However, a seller should only list an item to a particular site if he or she is willing to ship the item to that geographical reason. For example, a seller registered on the Germany eBay site may list items on the Germany and US sites, but should only do the latter if he is willing to ship the item to a buyer in the US.

Buyers typically log in to the global eBay site with which they are registered and so see only items listed on that site. If an item from another site is also listed on the site the buyer is logged into, they will see that item with some modifications. For instance, the user interface will be that of the buyer’s home site and in the associated language. But text that the seller specified for the item—such as the item title and description—appear in the language the seller used (eBay does not alter this information). Additionally, prices are converted to the currency of the buyer’s site. For example, if an eBay UK buyer is viewing an item that an eBay US seller listed to the UK site, that buyer will see prices converted to British Pounds—even though the seller listed the item in US Dollars. (This currency conversion is done automatically by eBay and is not something the seller or the seller application have to worry about doing.)

There are a number of special considerations an application must make for listing items to and retrieving items from sites other than the US. Not all features available on the US eBay site are

available on all global sites. See “[Field Differences for eBay Sites](#)” on page 590 for a breakdown of differences between global eBay sites.

The URLs below point to the various global sites that eBay currently hosts. As eBay continues to grow, so too will this list of sites. Please note that sites that do not contain “ebay” in the URL may not be accessible programmatically via the API. See “[Site IDs](#)” on page 611 for a current list of sites.

Argentina

<http://www.mercadolibre.com.ar>

Australia

<http://www.ebay.com.au/>

Austria

<http://www.ebay.at/>

Belgium

<http://www.ebay.be/>

Brazil

<http://www.mercadolivre.com.br/org-img/html/MLB/he.ml>

Canada

<http://www.ebay.ca/>

Canada (French)

<http://www.cafr.ebay.ca/>

France

<http://www.ebay.fr/>

Germany

<http://www.ebay.de/>

Hong Kong

<http://www.ebay.com.hk/>

Ireland

<http://www.ebay.ie>

India

<http://www.ebay.in>

Italy

<http://www.ebay.it/>

Korea

<http://www.auction.co.kr/default.htm?ssPageName=HKR01>

Malaysia

<http://www.ebay.com.my>

Mexico

<http://www.mercadolibre.com.mx/org-img/html/MLM/he.ml>

Netherlands<http://www.ebay.nl/>**New Zealand**<http://pages.ebay.com/nz/>**Phillipines**<http://www.ebay.ph>**Poland**<http://www.ebay.pl/>**Singapore**<http://www.ebay.com.sg/>**Spain**<http://www.ebay.es>**Switzerland**<http://www.ebay.ch/>**United Kingdom**<http://www.ebay.co.uk/>**United States**<http://www.ebay.com/>

PayPal

PayPal is the eBay online payment service. PayPal provides buyers with a safe and secure means to pay sellers for items they purchase. While PayPal is a separate site, it is closely integrated into the eBay selling and buying processes. Information about PayPal is available at the following locations:

PayPal Home Page<http://www.paypal.com/ebay>**eBay General Announcement Board**<http://www2.ebay.com/aw/marketing.shtml>

For more information on how PayPal works with eBay order line items and how to integrate PayPal into your applications, see “[Using PayPal](#)” on page 195.

eBay Motors

eBay Motors is a specialty eBay site on which vehicles and related items can be offered for sale. In the context of eBay Motors, vehicles includes cars, trucks, motorcycles, fire engines, tractors, boats, and even helicopters. In addition to vehicles, sellers can list vehicle parts and vehicle-centric memorabilia.

eBay Motors has a dedicated site ID, used when listing or viewing items. eBay Motors has a distinct category hierarchy in which items are listed and categories and their IDs are not the same as on any other eBay site. The eBay Motors category hierarchy needs to be retrieved by an application before that application could list or access items there, just as is the case for any one global eBay site.

There are special considerations an application must make when listing items to or retrieving items from eBay Motors. The most important of these considerations is the fact that the properties that define an eBay Motors item must be passed to eBay by using Item Specifics. For more information on listing an item to or retrieving an item from eBay Motors, see “[eBay Motors](#)” on page 559.

eBay Motors page
<http://pages.ebay.com/ebaymotors/index.html>

eBay Stores

eBay Stores is a specialty site on which selling businesses set up a distinct presence and buyers see only the items for that business.

Sellers can subscribe to be owners of an eBay Store (also referred to as a storefront). After they subscribe, they are known as store owners (store subscribers).

The seller can then customize this location and set up custom categories. Items that the seller lists are the only items that appear in the Store. This is as opposed to listing items in the general eBay categories, where a given category will have items from numerous sellers, all competing against one another for the attention of buyers.

An eBay Store enables a seller to set up a site with which the seller has an online presence dedicated to just that seller’s business. This makes it easier for a buyer to find items sold by that business, especially important once the business gains name recognition and cultivates a return business customer base.

An eBay Store owner can list items that only appear in that Store. The seller can also list auction items that appear both in the Store and in the general eBay categories.

There are special considerations an application must make when listing items to or retrieving items from eBay Stores storefronts. For more information on working with eBay Stores in applications, see “[Managing eBay Stores](#)” on page 534.

eBay Stores main page
<http://www.stores.ebay.com>

Shopping.com

The Shopping.com site is an eBay Site that allows you to compare products by price and consumer rating after narrowing your search using price range, volume, color, store, or other features, depending on the item’s category. You can identify sellers on the Shopping.com site by their Merchant name.

Shopping.com is a retail site, not an auction site:

<http://www.shopping.com/>

Half.com

The Half.com site enables you to sell new and used Books, CDs, Movies, Games, and Game Systems. The items you list for sale will be seen by millions of potential buyers. You can list Half.com items for longer durations than eBay auction and fixed-price listings. Half.com takes a small commission when your item sells (you do not pay fees for unsold items).

There are special considerations an application must make when listing items to or retrieving items from Half.com. For information on working with Half.com in applications, see “[Half.com](#)” on page 574.

Half.com by eBay
<http://www.half.ebay.com/>

Affiliate Tracking Concepts

Affiliates earn money from eBay for driving traffic to eBay. eBay knows that a sale came from your application because you include an affiliate ID in your API calls.

The following sections contain information about affiliate tracking:

“[Earn Commissions](#)” on page 155
“[Affiliate Tracking Enabled for Several Trading API Calls](#)” on page 156

Earn Commissions

Affiliate-related fields, which are included in a call request using the **AffiliateTrackingDetails** container, enable the tracking of user activity.

The **AffiliateTrackingDetails** container has fields such as **TrackingPartnerCode**, **TrackingID**, and **AffiliateUserID**.

The **TrackingPartnerCode** specifies the third party who is your tracking partner. It is required. Not all partners are valid for all sites. If you are registered with the eBay Partner Network, the **TrackingPartnerCode** is 9.

The **TrackingID** specifies an ID to identify you to your tracking partner. The value you specify is obtained from your tracking partner. If you are in the eBay Partner Network, the **TrackingID** is the Campaign ID ("campid") provided by the eBay Partner Network. A Campaign ID is a 10-digit, unique number. A Campaign ID is valid across all programs to which you have been accepted.

The **AffiliateUserID** need not be specified. You can define an **AffiliateUserID** (up to 256 characters) if you want to leverage it to better monitor your marketing efforts. If you are using the eBay Partner Network, and you provide an **AffiliateUserID**, the tracking URL returned by eBay Partner Network will contain the **AffiliateUserID**, but it will be referred to as a "customid".

The following example shows sample input with the **AffiliateTrackingDetails** container:

Example 7-2 Sample Input for Affiliate Tracking

```
...  
<AffiliateTrackingDetails>  
  <AffiliateUserID>234</AffiliateUserID>  
  <TrackingID>1234567899</TrackingID>  
  <TrackingPartnerCode>9</TrackingPartnerCode>  
</AffiliateTrackingDetails>  
...
```

When you use the **AffiliateTrackingDetails** container, a URL is returned that includes information for tracking user activity.

The following example shows a sample URL returned after you specified a **TrackingPartnerCode** of 9 in the **AffiliateTrackingDetails** container:

Example 7-3 Sample Output URL For Affiliate Tracking

```
http://rover.ebay.com/rover/1/711-53200-19255-0/  
1?campid=1234567899&customid=custominformation&toolid=10013&mpre=http://www.ebay.com
```

For information about the eBay Partner Network (eBay Affiliate Program), see the following location:

eBay Partner Network
<https://www.ebaypartnernetwork.com/>

Affiliate Tracking Enabled for Several Trading API Calls

For more information about the **AffiliateTrackingDetails** container, see the topics for the calls for which affiliate tracking is enabled:

PlaceOffer
<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/PlaceOffer.html>

FindProducts (Shopping API)
<http://developer.ebay.com/Devzone/shopping/docs/CallRef/FindProducts.html>

Part III

Selling Items on eBay

Categories	158
Listing Items	174
Using PayPal	195
Describing Items in a Listing	206
Using Description Templates	253
Multi-Variation Listings	264
Want It Now	309
Shipping	312
Specialty Categories and Listings	366

Chapter 8

Categories

On the eBay site, *categories* are a hierarchical set of groups in which items of a similar nature are listed. Listing an item in the proper category greatly increases the chances that an interested user will find and buy the item. Categories help buyers find the kinds of items they want, as the buyer only needs to look in one category or a few categories to find items they are interested in. Sellers benefit from the use of categories by the increased likelihood of a sale due to better and faster access to items by prospective buyers. For basic information on categories as basic building blocks, see “[Categories \(Introduction\)](#)” on page 138.

Each eBay site has its own set of categories. That is, the US eBay site has different categories than, say, the Germany eBay site. Many of the categories serve the same purpose on all sites (though they are named in the language of the site), but a site also contains categories that are only of regional interest or of interest to a particular culture.

To list an item on eBay or to search for items in a particular category, the user of your application needs to specify the numeric ID of the target category, and the category must exist on eBay. To help the user select a valid category ID, your application can download information about the category hierarchy (which includes category IDs and human-friendly names) and then present the data to the user.

Some listing features are only available in certain categories. For example, a seller can only specify a reserve price, offer Immediate Payment, or list with Item Specifics if the category supports such options. The category information you download from eBay includes meta-data about features that each category supports.

The remainder of this chapter discusses the means and strategies for retrieving the category meta-data for an eBay site, storing it for use in an application, using category information to support the listing process, and refreshing the locally stored meta-data. It is organized into these sections:

- “[Retrieving the Category Hierarchy for a Site](#)” on page 158
- “[Maintaining Category Data](#)” on page 161
- “[Determining Category Support for Certain Features](#)” on page 168
- “[Retrieving Suggested Categories Based on Keywords](#)” on page 171

Retrieving the Category Hierarchy for a Site

To retrieve the data for the categories on a particular site, use [GetCategories](#). Doing this involves three general steps: setting up the execution environment, specifying the data to return (category hierarchy version or the category hierarchy itself), and making the API call.

The ID for a category (one of the values returned by [GetCategories](#)) is one of the required inputs for listing a new item.

After retrieving the data for a site's category hierarchy, the application should store the information in some persistent form, such as in a database table. This data can then later be refreshed when a category hierarchy change is indicated (i.e., when the returned category hierarchy version is different than the last one the application retrieved).

Optionally, specify the eBay site for which to retrieve the category data. The category data varies for different sites. That is, the category hierarchy on the eBay US site is not the same as the category hierarchy on the eBay Germany site. If the category site ID is not specified, eBay will return category data for the site you specify in the request URL (see “[Routing the Request \(Gateway URLs\)](#)” on page 87).

Optionally specify the ID of a parent category to be returned as the top most category in the result set. If not specified, then the top most category returned is the root category for the target site. If a parent category is specified, then that is the top most category in the result set and only subcategories child to it (and their children) are returned. To determine the available category parent IDs, retrieve all category data the first time you execute [GetCategories](#), and store the data. This may take several minutes. Subsequently, refer to your locally stored data for specific parent category IDs.

The depth of a category within the hierarchy is called a *level*. The top-most categories have a level of 1, their immediate children have a level of 2, and so on. Optionally, specify a limit on the number of levels of the category hierarchy to retrieve. If no level limit is specified, the result set will contain categories at all levels. If a level limit is specified, then only those categories having that depth level or less are returned. For example, if a value of 1 is specified, then only those categories child to the root category are returned. The categories that are grandchildren to the root category are not returned.

Optionally specify whether to retrieve all categories or only categories in which items can be listed (*leaf* categories). If you want to enable users to browse a hierarchical category structure to choose a category to list in, you should retrieve all categories. If you want to verify which leaf categories support a particular feature, you can retrieve only the leaf categories.

Table 8-1 GetCategories Data Specifiers

Input Field	Required?	Type/Code List	Purpose
CategorySiteID	Y	string	Specifies the eBay site from which the call will retrieve the category data. Pass a string containing the numeric ID that identifies the site (e.g., “0” for US). See “ Site IDs ” on page 611.
CategoryParent	N	string	Specifies the category ID for the top most category to return (along with the subcategories under it, the value of the LevelLimit property determining how deep). May specify multiple top most categories and the hierarchies for each is returned.
LevelLimit	N	int	Specifies which category levels to return. Pass an int value as its input to retrieve all category nodes with a category level less than or equal to the LevelLimit value.
ViewAllNodes	N	boolean	If you specify a false value, only leaf categories are returned. If you specify a true value, all categories are returned.

The example below shows retrieving category data for the US eBay site. It requests categories up to three levels deep, returning both leaf and parent categories.

Example 8-1 Specifying the Category Data

```
<?xml version="1.0" encoding="utf-8"?>
<GetCategoriesRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <RequesterCredentials>
        <eBayAuthToken>ABC...123</eBayAuthToken>
    </RequesterCredentials>
    <CategorySiteID>0</CategorySiteID>
    <DetailLevel>ReturnAll</DetailLevel>
    <LevelLimit>3</LevelLimit>
    <ViewAllNodes>true</ViewAllNodes>
</GetCategoriesRequest>
```

GetCategories returns in the response object a **CategoryArray** containing one or multiple (typically many) **CategoryType** elements, one for each of the categories on the specified eBay site.

Table 8-2 GetCategories Return Values

Return Fields	Type/Code List	Purpose
CategoryArray	CategoryArrayType	List of the categories from the specified site (possibly limited to only those of a certain level, based on input). Contains one CategoryType object for each returned category, each of which contains the data for one category.
CategoryCount	int	Indicates the number of categories returned by the call.
UpdateTime	dateTime	Indicates the date and time the just-retrieved category hierarchy was last modified.
CategoryVersion	string	Indicates the update version for the just-retrieved category hierarchy.

An application needs to traverse the **Category** elements in the **CategoryArray** property of the response XML to visit each one and inspect its properties in order to store and work with the data in a useful way.

Each **Category** element returned has the properties for a category, such as the category ID, category name, its parent ID (if any), whether or not it is a leaf category in which items can be listed, and other information.

Category IDs are a required input for listing a new item (except Store Inventory items). The category names can be used by an application to populate a list in its user interface, from which the user can select the category in which to list a new item. The application should store the category data in some persistent manner (such as a database table) to have the data available whenever the application is run.

Different leaf categories can have the same names but different IDs. For example, under the general category of “Books”, several leaf categories might exist called “Fantasy”. In some cases, this is because the nature of the item is different (e.g., an audio book on CD vs. cassette). In other cases, it is

because sellers may prefer to list items in particular parent categories in order to target buyers more carefully. Some examples of leaf categories with identical names might be:

- Audio: Cassettes: Fiction: Fantasy
- Audio: CDs: Fiction: Fantasy
- Children: Fiction (Chapter Books): Early Readers: Fantasy
- Children: Fiction (Chapter Books): Middle School: Fantasy
- Fiction & Literature: Romance: Fantasy

All categories except those at the top level will have a parent category, and thus a non-zero value in **CategoryParentID**. Categories for which the **LeafCategory** value is true are at the bottom-most level of their branch in the hierarchy and have no child categories.

The combination of the **CategoryID**, **CategoryParentID**, and **LeafCategory** provide the information you need in order to represent the category hierarchy. With these values, you can traverse the hierarchy and enable users to drill down from higher-level categories to lower-level categories.

For example, an application might initially present a list of categories that are all at the same level. These are categories that all have the same **CategoryLevel** values. Of these categories, some will be container categories (i.e., the **LeafCategory** value is `false`). To drill down into the next level of one branch of the hierarchy, the application would use the **CategoryID** of one of those categories and then load all categories where the **CategoryParentID** is the same as that **CategoryID** (essentially loading all of that category's children). To drill down to the bottom of the category branch, this process can be repeated as many times as necessary until all the child categories are leaf categories (i.e., the **LeafCategory** value is `true`).

These fields can also be used to “walk” upwards in the hierarchy. To do that, find the category for which the **CategoryID** equals the **CategoryParentID** of the current category (i.e., find the current category's parent). To find sibling categories, find all of the parent category's child categories (those categories for which the **CategoryParentID** is the same as the **CategoryID** of the current category). You can use this information to design and present category-selection functionality in your application.

GetCategories also returns the date and time the category hierarchy was last updated and the version of the category hierarchy.

See “[Maintaining Category Data](#)” on page 161 for information about how to refresh the locally stored category meta-data after it has been initially downloaded.

Maintaining Category Data

eBay periodically adjusts a site's category hierarchy to meet the current needs of eBay members and the marketplace. These kinds of changes are intended to make it easier for buyers to find items they

are seeking and to help sellers list their items where buyers are most likely to look for them. For example, eBay can make the following kinds of category changes:

- Add a new category for a new kind of item.
- Split an existing category into multiple categories to help users differentiate items.
- Combine multiple categories into a single category to make very similar items easier to find.
- Allow a category that is rarely used to expire without being combined with other categories.
- Add new meta-data about the features a category supports (e.g., reserve prices).

This means the available list of categories changes periodically. After retrieving a site's category data, your application should store the information in some persistent form, such as in a database table. Some category IDs can change as part of this process. This means that when you submit a listing to eBay, it is possible that the category ID will no longer exist on eBay and the listing request will fail. (Similarly, if you pass an old category ID in a search request, the request may fail.)

Periodically, your application should refresh the locally stored category data by synchronizing it with the latest meta-data on the eBay site. This is an important part of making sure a seller's listing data is valid.

Here are some typical cases in which even a new listing's category information might be outdated:

- Your application's locally stored category meta-data was out of date when the seller defined the listing. This might occur with a client application that allows a user to define listings offline without first making sure their locally stored meta-data is synchronized with eBay's data.
- Your application supports the ability to take an existing (previously defined) item and sell a similar item. For example, if the seller sold an item six months ago and now wants to sell a similar one, the application retrieves the old item's details from a local database and uses the information to pre-fill the details of a new listing. However, the old item's category and other data may no longer be valid on the site.
- Your application allows a user to create a listing in advance of the start date and then schedule the listing to become active at a later date (often called a *pending* or *scheduled* listing). Even if the locally stored meta-data is synchronized with eBay's data at the time the listing is defined, the listing details may be out of date by the time the application submits the information to eBay.

Note: The easiest way to create scheduled listings and ensure that their details will be valid when the listing date arrives is to use eBay's built-in listing scheduling functionality. Instead of creating the listing locally and submitting it later, you specify the listing start date in the `Item.ScheduleTime` property and submit the item immediately. When the scheduled time arrives, eBay updates the category and other information for you and the listing becomes active. This means no additional validation is required on your part – even if the category ID or other metadata has changed on the site.

The sections below describe some of the ways you can keep your locally stored category information up to date.

When eBay combines a category with another category (or splits a category into multiple categories), we move any active and scheduled items from the old category into the appropriate active category. In this case, we update the category information stored on the items. This is necessary to make sure

that the items will appear when users browse and search for items. For example, this ensures that [GetItem](#), and related calls always return the item's current category.

Thus, it can also be useful to synchronize listings that have been submitted by your users, to ensure that the categories and other data of items you have stored locally match the data that is currently in use for those items on the eBay site. Use calls like [GetSellerList](#) and [GetSellerTransactions](#) to perform such tasks.

Checking the Category Hierarchy Version

The category hierarchy for a given site is versioned. The first time your application retrieves the categories for a particular site, use a detail level value of [ReturnAll](#). This returns the category data and the version for that site's category hierarchy. Store this value for future use.

When the application next seeks to refresh the category data for that eBay site, call [GetCategories](#) with no detail level specified. This causes [GetCategories](#) to return only the category hierarchy version. Compare this newly retrieved version number with the one retrieved in the last call. If the newly retrieved version number is different, call [GetCategories](#) again but with a detail level of [ReturnAll](#) to retrieve the category data. Store the new version number as the reference point for the next call. If the new and stored version numbers are the same, then category data does not need to be retrieved for that site at that time.

Note that changes to the category hierarchy of one site are independent of changes to another site. So while one site's categories may not have changed, the categories for other sites may have. The category hierarchy version must be checked separately for each eBay site the application supports, using the procedure described here.

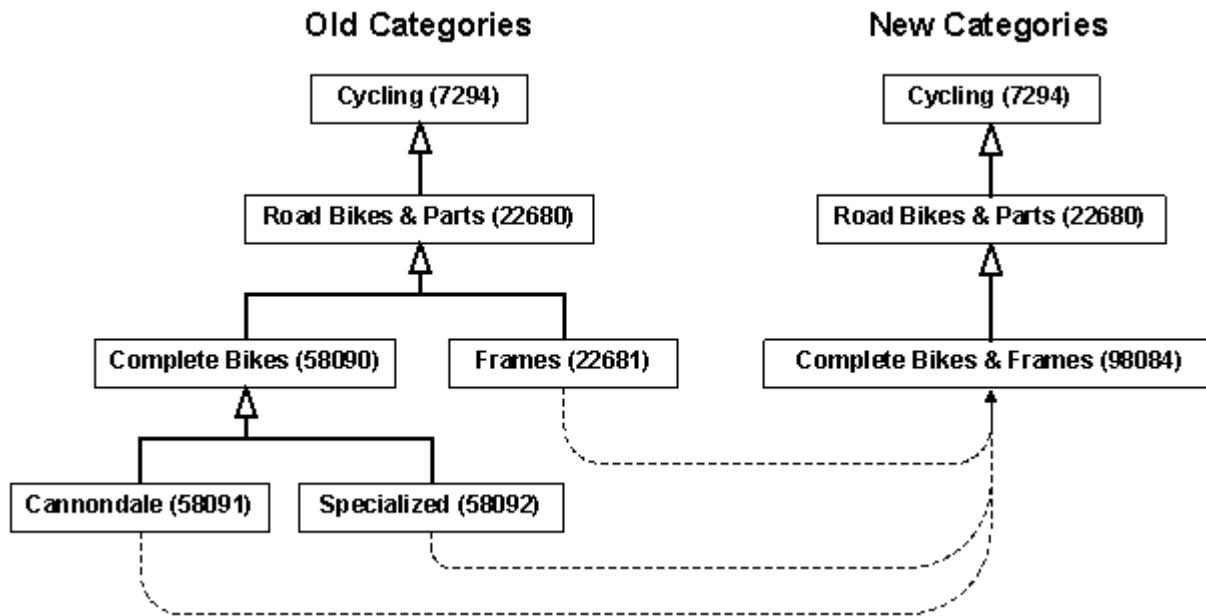
The eBay category hierarchy is updated on a monthly basis, but may be updated more frequently than this interval. It is a good idea to use [GetCategories](#) to check the category tree version at least once on each day that the application is being used.

Mapping Old Category IDs to Current IDs

The ID for a category in which you list an item can change. eBay might split categories to help users distinguish between items more easily or combine multiple categories to make very similar items easier to find. (In some cases, eBay may allow a category that is rarely used to expire without being combined with other categories.)

[Figure 8-1](#) shows an example of three leaf categories (Frames, Cannondale, and Specialized) that were combined into a single leaf category (Complete Bikes & Frames).

Figure 8-1 Old Categories Combined Into an Active Categories



A common way to determine whether a category ID is current is to make sure your locally stored category meta-data is current (see “[Checking the Category Hierarchy Version](#)” on page 163) and then verify that the ID exists in the current meta-data. (To be valid in listing requests, the category ID also must be a leaf category.) In certain cases, even an item your application recently defined can have an old category ID (see “[Maintaining Category Data](#)” on page 161 for examples).

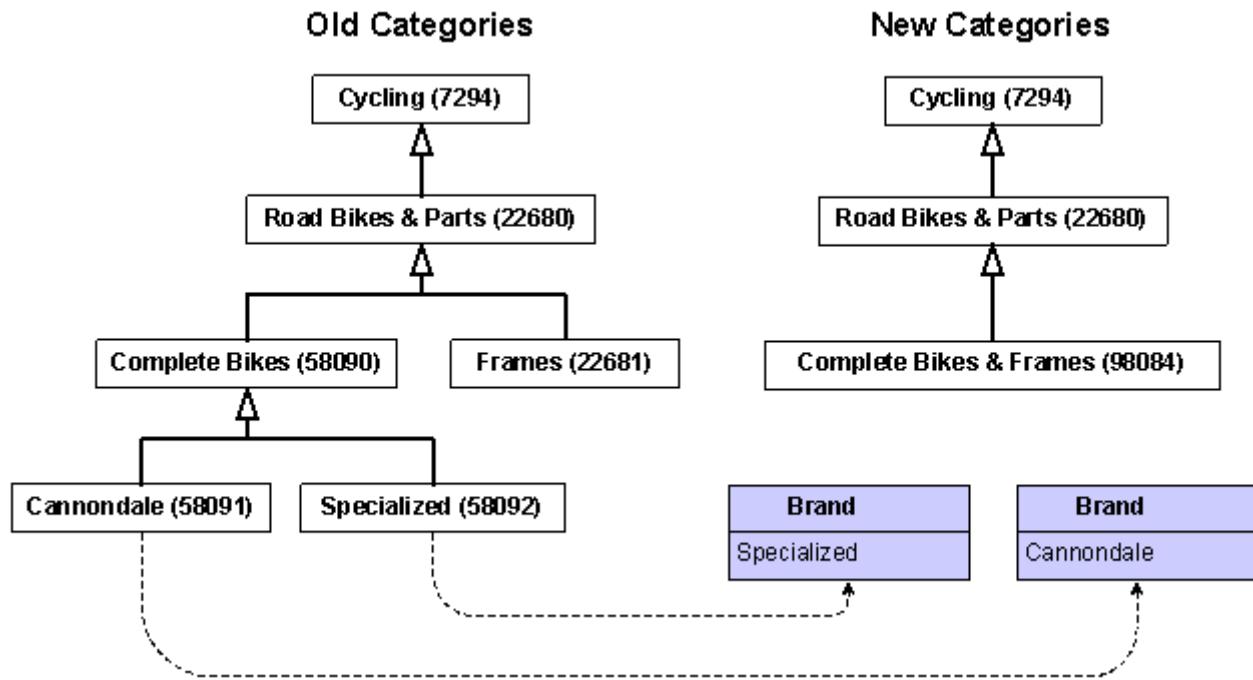
If a seller wants to list an item that was defined with an old category ID, you can handle the situation in different ways:

- Allow eBay to attempt to update the category for you. This approach may be more convenient for sellers who are listing hundreds of items at once. In some cases, invalid data may be dropped from the listing, but the listing will still be submitted successfully. See “[Using eBay’s Category Mapping Feature](#)” on page 165 for information about this approach.
- Update the category information on the client side. This approach may be more convenient for sellers who use listing features that are highly dependent on the category (such as reserve prices) or who prefer not to allow eBay to drop invalid data. See “[Mapping Categories on the Client Side](#)” on page 166.

If the old categories were combined into a more general category, the original differentiation that the old categories provided may be lost. For example, the difference between “Cannondale” and “Specialized” would no longer be evident from the new “Complete Bikes & Frames” category. If the new category supports Item Specifics, eBay may offer Item Specifics that convey the same details as the original categories. In such cases, eBay may provide the ability to pre-fill these Item Specifics based on the old category.

[Figure 8-1](#) shows an example of the way eBay uses old leaf categories (Cannondale and Specialized) to pre-fill corresponding Item Specifics (Brand=Cannondale and Brand=Specialized).

Figure 8-2 Item Specifics Pre-filled Based on Old Categories



If a seller wants to include Item Specifics that correspond to old category IDs, you can handle the situation in different ways:

- Allow eBay to attempt to pre-fill the Item Specifics based on the category (or categories). This approach may be more convenient for sellers who are listing hundreds of items at once. See ["Using eBay's Category Mapping Feature"](#) on page 165 for information about this approach.
- Update the Item Specifics information on the client side. This approach may be more convenient for sellers who want more control over the Item Specifics that are selected. See ["Mapping Categories on the Client Side"](#) on page 166.

Using eBay's Category Mapping Feature

When you list an item, you can use the `Item.CategoryMappingAllowed` flag to allow eBay to handle the cases in which the primary or secondary category ID specified in the request is no longer valid. When set to `true`, this flag allows eBay to check whether you are passing in old category IDs and, if you are, to map them to corresponding active IDs (if any exist). This enables listing requests with old categories to succeed without category validation errors. High-volume sellers may find it more convenient to submit listings in this manner instead of first updating all their categories each time.

In some cases, it is possible that when eBay maps an item's old category to an active category, eBay will drop certain data that is invalid for the active category. Therefore, it is a good idea to let the seller decide (through a standard preference or for each listing) whether or not to allow eBay to map a listing's categories in this manner during the listing flow.

To use this category mapping feature:

- 1 Set the `Item.CategoryMappingAllowed` flag to true.

If a category ID passed in the request longer exists, eBay maps the old ID to the new ID and uses the new one for the listing instead. eBay returns the new ID in the `CategoryID` element (and `Category2ID`, or both, as appropriate) in the response.

If the category ID has not changed, no mapping occurs and the `CategoryID` (or `Category2ID`) element is not returned in the response. This means you can always specify `CategoryMappingAllowed` in listing requests even when you believe the item's categories are current.

- 2 If category mapping occurs, update the category data on the locally stored item so that it is consistent with the item stored on eBay. Also make sure your application's locally stored category metadata is current (see "[Checking the Category Hierarchy Version](#)" on page 163).
- 3 It may also be possible to use Item Specifics to convey the same differentiation as the original category did. Let the seller decide whether or not to allow eBay to attempt to map the category to an item specific. To use this feature, set the `Item.CategoryBasedAttributesPrefill` flag to `true`.

If the category ID passed in the request has been mapped to an item specific, eBay includes the item specific in the listing's Item Specifics. See "[Mapping Old Category IDs to Current IDs](#)" on page 163 for more information and other options.

If the seller prefers not to allow eBay to perform category mapping on their behalf, see "[Mapping Categories on the Client Side](#)" on page 166.

If you prefer not to map categories locally and the seller chooses not to allow eBay to perform category mapping:

- 1 Set the `Item.CategoryMappingAllowed` flag to false (or do not specify it at all).
If a category ID passed in the request no longer exists, the call returns an error.
- 2 If the call returns an error, make sure your application's locally stored category meta-data is current (see "[Checking the Category Hierarchy Version](#)" on page 163) and then prompt the seller to choose an active category.

The `CategoryMappingAllowed` setting is transient. That is, it only applies for the current call; it is not stored with the item data. This means that when you revise or relist an item, it will not retain the originally category mapping preference that you specified in `AddItem`.

Some categories can expire without being combined with other categories. If the category ID you pass in `AddItem` or related calls has expired in this manner, the `CategoryMappingAllowed` flag will not have any effect and an error will indicate that the category is invalid.

Mapping Categories on the Client Side

In some situations, you may need to update an item's category information on the client side before you list the item. For example, this approach may be more convenient for sellers who use listing features that are highly dependent on the category (such as Item Specifics) or who prefer not to allow eBay to drop invalid data.

Before listing an item, compare its primary and secondary category IDs to the current category meta-data. Make sure your application's locally stored category meta-data is up to date (see "Checking the Category Hierarchy Version" on page 163).

If the item's category data is no longer current, prompt the user to choose a new category. In this case, you can suggest the active leaf category (if any) that is equivalent to the old category the seller originally chose. Alternatively, you can update the item's category information without prompting the user. Some applications allow the user to set a preference indicating whether to allow this.

To determine which active category has replaced an old category, use [GetCategoryMappings](#) (see "GetCategoryMappings" on page 810). The response provides a map between categories that are no longer active and the corresponding active categories. Multiple old categories can be mapped to the same active category. See [Figure 8-1](#) on page 164 for an illustration of categories that have been combined.

Note: [GetCategoryMappings](#) only returns information about categories that have been mapped (i.e., combined categories and split categories). It does not return information about expired categories that have no corresponding active categories. When a category expires in this manner, any completed items that were listed in the expired category can still be found, but new listings cannot be created in the category.

Like the active category hierarchy, the category mappings vary for each eBay site and the data is versioned. For a given site, you can reduce the amount of data returned by only retrieving data that has changed since the last time you executed the call. You can also reduce the number of times you execute the call by using certain practices:

- 1 Initially, use [GetCategoryMappings](#) with a [DetailLevel](#) value of [ReturnAll](#) and no [CategoryVersion](#) to get all category mappings. Save the category mapping data locally, including the version value that is returned. This step should be performed at the same time you execute [GetCategories](#) so that the metadata returned from these calls is consistent.
- 2 Mappings only change when the category hierarchy is updated. If the version returned from [GetCategories](#) does not change, you can assume the category mapping metadata also has not changed. That is, it is not necessary to execute [GetCategoryMappings](#) when the [GetCategories](#) response has not changed.

Alternatively, check whether the mappings have changed by using [GetCategoryMappings](#) with no [DetailLevel](#) value. This returns the current version information only (with no mappings). Compare the version with the last saved mapping version value to see if the mapping data has changed. If the version has not changed, it is not necessary to retrieve the mappings.

- 3 When the category hierarchy changes (or you determine that the mapping version has changed), use [GetCategoryMappings](#) with a [DetailLevel](#) value of [ReturnAll](#) and specify the last saved mapping version value in the [CategoryVersion](#) input field. This limits the response to new and changed mappings since that version. Store the new [CategoryVersion](#) value returned and use it in the same manner the next time you execute this call.

[GetCategoryMappings](#) returns old and active category IDs but it does not distinguish between parent and leaf categories or return category names. Therefore, combine the results with information from [GetCategories](#) and related calls in order to build a complete set of information about each category. That is, match the active category IDs returned from [GetCategories](#) to the active category IDs returned from [GetCategoryMappings](#) in order to determine whether the old categories have been mapped to leaf categories or to display an active category name to the user.

Use the category mapping information to update category IDs defined on items that you plan to list on eBay.

[GetCategoryMappings](#) returns current mappings only (not historical mappings). For example, the mapping of a category could occur in this manner:

- 1 In January, eBay maps category 123 to category 456.

Thus, in January the 123-to-456 mapping is returned.

- 2 In May, eBay maps category 456 to category 789.

Category mappings are transitive, so in May category 123 is now also mapped to category 789.

Thus, in May two mappings are returned:

- One for the 123-to-789 mapping
- One for the 456-to-789 mapping.

The previous 123-to-456 mapping is no longer returned because 456 is no longer an active category.

This logic is also illustrated in the table below.

Table 8-3 Category Mappings Over Time

Mappings Returned	Old ID	Active ID
January	123	456
May	123	789
May	456	789

Note: The category IDs in the above example are fictitious and are only used to illustrate the transitive nature of mappings. These values may or may not correspond to actual category IDs on an eBay site.

When eBay combines an old leaf category (e.g., Cannondale) with a more generic leaf category (e.g., Complete Bikes & Frames), eBay may instead use item specifics for the data that corresponds to the old category (e.g., Brand=Cannondale). Use [GetCategorySpecifics](#) to determine which item specifics might be available for the new category.

Determining Category Support for Certain Features

As described earlier, [GetCategories](#) returns the core information about each category, such as the category name, ID, level in the hierarchy, and so forth. It also returns information about each category's parent ID so that you can build the category tree within your application.

[GetCategories](#) also returns information about listing features that are only supported by individual categories. For example, a seller can only specify a reserve price if the category supports reserve prices.

As the volume of the data returned from [GetCategories](#) is large, information about a site's support for certain additional features is returned by [GetCategoryFeatures](#) (see “[GetCategoryFeatures](#)” on page 808).

To determine which categories support listing or searching with Item Specifics, use [GetCategoryFeatures](#) and [GetCategorySpecifics](#).

When you use calls like [GetCategoryFeatures](#), you need to combine the results with information from [GetCategories](#) in order to build a complete set of information about an individual category and the features it supports. For example, [GetCategoryFeatures](#) returns category IDs but it does not distinguish between parent and leaf categories. Therefore, you need to match the category IDs returned from [GetCategories](#) to the category IDs returned from [GetCategoryFeatures](#) in order to determine which leaf categories support Item Specifics.

The following sections describe how to use these calls to determine which sites or categories provide support for particular features.

Note: The category meta-data might not include all features that are restricted to certain categories. Refer to the documentation of each feature you are using to determine if restrictions exist that cannot be determined programmatically.

Determining Whether a Category Supports Reserve Prices

A category can allow listings with reserve prices. A reserve price is the lowest price at which a seller is willing to part with the listed item.

Some categories that allow listings with reserve prices also allow you to reduce those reserve prices when you modify an item. Other categories do not.

An entire site may or may not support reserve prices on items being listed. This value is indicated by the [ReservePriceAllowed](#) field returned by [GetCategories](#). If [ReservePriceAllowed](#) is `true`, the site overall allows reserve prices. Note that [GetCategories](#) returns the value of [ReservePriceAllowed](#), but you cannot set it.

Similarly, an entire site may or may not support reducing reserve prices on listed items. This value is indicated by the [ReduceReserveAllowed](#) field returned by [GetCategories](#). If [ReduceReserveAllowed](#) is `true`, the site overall allows reducing reserve prices. Note that [GetCategories](#) returns the value of [ReduceReserveAllowed](#), but you cannot set it.

[GetCategories](#) also returns the value of [MinimumReservePrice](#), the lowest reserve price allowed for the site, but does not set it.

The [ORPA](#) (Override Reserve Price Allowed) field in [Category](#), if set to `true`, reverses the site's [ReservePriceAllowed](#) setting for the category. That is, if [ReservePriceAllowed](#) is `false` for the site and [ORPA](#) is `true` for the category, the site does not allow reserve prices in general but the category does allow reserve prices. If [ReservePriceAllowed](#) is `true` for the site and [ORPA](#) is `true` for the category, the site allows reserve prices in general but the category does not allow reserve prices.

You cannot set the [ORPA](#) field for a category. Categories are defined for individual eBay sites by the eBay site administrator and vary from one site to the next. That is, the US eBay site has different categories than, say, the Germany eBay site.

You can, however, use [GetCategories](#) to return an array of categories and parse the response for the values of [ReservePriceAllowed](#), [MinimumReservePrice](#), and [ORPA](#) to determine the rules for a given category. Any combination of values for the three fields is possible.

Similarly, the ORRA (Override Reduce Reserve Allowed) field in [Category](#), if set to `true`, reverses the site's [ReduceReserveAllowed](#) setting for the category. That is, if [ReduceReserveAllowed](#) is `false` for the site and [ORPA](#) is `true` for the category, the site does not allow sellers to reduce reserve prices in general but the category does allow sellers to reduce reserve prices. If [ReduceReserveAllowed](#) is `true` for the site and [ORPA](#) is `true` for the category, the site allows sellers to reduce reserve prices in general but the category does not allow sellers to reduce reserve prices. You cannot set the [ORRA](#) field for a category. Parse the response for the values of [ReduceReserveAllowed](#), [MinimumReservePrice](#), and [ORRA](#) to determine the rules for a given category.

Depending on the site and category, you also may need to call [GetCategoryFeatures](#) and check [MinimumReservePrice](#).

Remember that:

- [ReservePriceAllowed](#) and [ReduceReserveAllowed](#) apply to the site overall.
- The [ORPA](#) setting defined in [Category](#) reverses the value of [ReservePriceAllowed](#) for that category.
- The [ORRA](#) setting defined in [Category](#) reverses the value of [ReduceReserveAllowed](#) for that category.
- [MinimumReservePrice](#) defines the lowest reserve price allowed for the site overall. If the category or site does not allow reserve prices, it will be ignored.

Determining Whether a Category Supports Item Specifics

See “[Retrieving Category-to-Meta-Data Mappings](#)” on page 563.

Determining Whether a Category Supports Product Details

See “[Retrieving Category-to-Meta-Data Mappings](#)” on page 563.

URLs for Category Lists for Global eBay Sites

The table below shows the URLs for the category lists on the various world-wide eBay sites.

Category numbers are currently not available for the following sites: HK, MY, PH, PL, SE, SG.

See also “[Site IDs](#)” on page 611.

Table 8-4 Category Lists for Global eBay Sites

Site	URL
AT (16)	http://listings.ebay.at/
AU (15)	http://listings.ebay.com.au/

Table 8-4 Category Lists for Global eBay Sites (Continued)

Site	URL
BEFR (23)	http://listings.befr.ebay.be
BENL (123)	http://listings.benl.ebay.be
CA (2)	http://listings.ebay.ca/
CAF'R (210)	http://listings.cafr.ebay.ca/
CH (193)	http://listings.ebay.ch
DE (77)	http://listings.ebay.de/
ES (186)	http://listings.ebay.es/
FR (71)	http://listings.fr.ebay.com/
HK (201)	http://listings.ebay.com.hk/
IE (205)	http://listings.ebay.ie/
IN (203)	http://listings.ebay.in
IT (101)	http://listings.ebay.it/
MY (207)	http://listings.ebay.com.my
NL (146)	http://listings.ebay.nl/
PH (211)	http://listings.ebay.ph/
PL (212)	http://listings.ebay.pl/
SG (216)	http://listings.ebay.com.sg
UK (3)	http://listings.ebay.co.uk/
US (0)	http://listings.ebay.com/
eBay Motors (100)	http://listings.ebaymotors.com/

Retrieving Suggested Categories Based on Keywords

Sellers often want to know where other sellers are listing items similar to theirs.

[GetSuggestedCategories](#) returns a list of the categories with the highest number of items whose titles or descriptions contain keywords you specify in a query. The list contains up to 10 categories and shows the percentage of matches that fall into each category.

Categories are arranged in a hierarchy that can be several levels deep. A given category can be either a leaf node (in which items can be listed) or a parent node (containing other parent or leaf nodes). All of the categories that [GetSuggestedCategories](#) returns are leaf nodes. For each leaf node, [GetSuggestedCategories](#) returns the leaf category name and information to help you build the fully qualified name. The leaf category name is returned in the [CategoryName](#) field, in a form like "Architecture" (without the quotes). The parent category names are returned in an array of one or more [CategoryParentName](#) fields, in hierarchical order.

Example 8-2 shows the structure for the Art:Prints:Antique (Pre-1900):Architecture category.

Example 8-2 A Fully Qualified Category

```
<SuggestedCategory>
  <Category>
    <CategoryID>10165</CategoryID>
    <CategoryName>Architecture</CategoryName>
    <CategoryParentID>550</CategoryParentID>
    <CategoryParentID>20140</CategoryParentID>
    <CategoryParentID>10163</CategoryParentID>
    <CategoryParentName>Art</CategoryParentName>
    <CategoryParentName>Prints</CategoryParentName>
    <CategoryParentName>Antique (Pre-1900)</CategoryParentName>
  </Category>
  <PercentItemFound>4</PercentItemFound>
</SuggestedCategory>
```

GetSuggestedCategories also returns a leaf category ID and the parent IDs in the same manner.

Your application can display the full category name or ID to help a user distinguish between leaf categories. However, when you use AddItem and other related calls, use only the leaf category name or ID.

A seller can determine which categories contain the most matching items by comparing the values returned in the PercentItemFound fields. PercentItemFound indicates what percentage of the matching listings fall into each category. The total of percentages across the categories may be less than or greater than 100%. When you look at the percentages, remember that:

- They are rounded to the nearest whole number.
- Categories with percentages that round to 0% are not returned.
- Only the categories with the highest quantity of matching items are returned.
- The maximum number of categories returned is 10.
- Items that are listed in two categories count towards both of those categories.

The total of values in PercentItemFound across the categories may be less than or greater than 100%.

Specifying a Query for Suggested Categories

The query, defined by the **Query** field, is a search string. The search string consists of one or more keywords to search for in the listing title. The words *and* and *or* are treated like any other word. You can use boolean AND and OR logic by using certain modifiers instead.

eBay is deprecating wildcard (*) queries from all search tools and functions, including Web and mobile search, saved searches, and searches made through API calls. For more information, refer to the Wildcards in eBay Searches deprecated product update page.

To retrieve suggested categories using the API:

- Define the endpoint to the eBay sandbox or production server and create the query string.
- Authenticate to the server with a valid token.
- Create service locator and service objects.
- Create a [GetSuggestedCategoriesRequestType](#) object.
- Set the query in the request object.
- Create a [GetSuggestedCategoriesResponseType](#) object.
- Use the service object to call [GetSuggestedCategories](#), passing it the request object and returning results to the response object.
- In the response, get the [SuggestedCategoryArrayType](#) and parse it to retrieve the values you want.

Chapter 9

Listing Items

- “Listing Policies” on page 174
- “Listing an Item” on page 176
- “Specifying a Payment Method” on page 184
- “Requiring Immediate Payment” on page 187
- “Fees Resulting from Listing an Item” on page 189

The primary force that drives the eBay model is the sale of things to buyers. The first step in selling to buyers is listing an item on the eBay site.

An item can be listed with the eBay user interface, which is referred to as the Sell Your Item pages (or SYI for short). An item can also be listed using an application.

Item listings created on an eBay site and through an application are functionally the same. When an item created using an application is viewed on the eBay site, it looks and behaves the same as an item listed using the eBay site. The reverse is also true.

In the Trading API, an item listing is represented by the **Item** object. There are a number of aspects to an item, from the title and price, to the amount of time the listing appears on eBay, to how the item appears visually. These aspects translate directly to properties of the **Item** object. So when an application lists an item with values specified by the seller, it is closely analogous to the seller supplying the same values on the SYI page.

Some item properties affect the appearance of the listing on various pages where items are viewed (the View Item page, the search page, the category browse page). Other properties affect how a prospective buyer goes about purchasing the item and how much that buyer needs to pay. Still other properties do not have an effect until later on, in the checkout process.

Listing Policies

eBay’s Listing Policies
<http://pages.ebay.com/help/policies/listing-ov.html>

How eBay Protects Intellectual Property (VeRO)
<http://pages.ebay.com/help/tp/programs-vero-ov.html>

Seller Limits

As of October 2010 on the US eBay site, *seller limits* are in effect for new sellers. Users registering as sellers on or after the effective date are subject to these initial selling limits; existing sellers are currently not affected. These seller limits restrict the new seller to a maximum quantity of items per month or to a maximum value of items listed per month, whichever limit is reached first. The actual limits can vary per seller, depending on various factors.

One purpose of the seller limits is to improve the buyer experience. The limits imposed on new, inexperienced sellers are intended to let the seller gain needed experience and limit any mistakes to a small number of items and their buyers. As the seller gains experience, the seller can request higher limits.

Certain sellers may be new to eBay, but are already large, established sellers outside of eBay. Sellers of this type might want to try requesting higher initial limits from eBay. (Any seller can try to increase his/her limit at any time using the **Request higher selling limits** link in the All Selling section of My eBay.)

What Applications Need to Do

An application that lists items needs to provide appropriate messaging to sellers who are subject to the seller limits. For sellers who have a strong business case for higher initial limits, the application should encourage them to apply for higher limits immediately by using the **Request higher selling limits** link in the All Selling section of My eBay under the Selling Limits heading. The link takes the seller to various options for increasing limits, and the seller can choose between these.

The listing application messaging must make it clear that selling limits exist for the seller and also keep the seller informed about how close the seller is to the limits. The application could optionally make listing suggestions based on the value and number of the items the seller is attempting to list; for example, your application could track the listings being entered and inform the seller as the seller approaches the limits. An application could also make suggestions as to the items that are most likely to sell, and so forth.

The key API component you use for this messaging is GetMyeBaySelling, which returns the remaining quantity of items that the seller can sell, and the remaining value amount, via the **Summary.AmountLimitRemaining** and **Summary.QuantityLimitRemaining** elements (You need to set **SellingSummary.Include** to true in the request.)

More Details on How Limits Work from Perspective of the eBay UI

Sellers are limited in the number of sold items or active listings they can have in any month. They are also limited in the total value of the items they can sell in any month. If an item sells, it counts toward the limit, but if a listing ends without any sold items, it no longer counts against the limit.

For example, if your limit is 100 sold items or active listings per month, and you create a fixed price listing for 100 items, you've reached your limit. But if that listing ends 7 days later and you've sold 10 of the 100 items, you can list 90 more items that month. You can continue until you've sold 100 items. Once you reach your limit, you have to wait until the beginning of the next month to list more items.

This limit is in addition to a limit on the total value of the items you can sell in a month. For example, if your limit is 100 sold items or active listings, or \$1,000 in total value sold in a month, and you sell 10 items for \$100 each, you have reached your limit.

Listings that carry over from one month to the next also apply to this limit. For example, if you list 100 Good 'Til Cancelled items on May 1, and by May 31, 50 of them have sold, you can only list another 50 items between June 1 and June 30, unless you cancel some of the 50 active Good 'Til Cancelled listings.

Finding out your limits

You can find your overall limits in My eBay. Click My eBay at the top of most eBay pages, sign in to your account, and click the Sell link on the left side of the page.

You'll see a red alert at the bottom of the Monthly selling limits section once you've reached a limit, letting you know you won't be able to sell more items in the current month but giving you an option to increase your limits.

Categories Exempt from Limits

Certain categories are not subject to seller limits:

- Business & Industrial
- Real Estate
- Motor Vehicles (except for Parts & Accessories, which are subject to seller limits)
- Classified Ad

Requesting a limit increase

As you get close to or meet one of these limits, you'll see a message when you list an item letting you know. The message will give you additional instructions on requesting a limit increase. You'll also see the **Request higher selling limits** link in the All Selling section of My eBay under the Selling Limits heading, which will take you to a page flow where you can choose among the options presented for requesting higher limits.

Listing an Item

This section describes how to list an item.

Note: `GetMyeBaySelling` returns seller limit information for the seller if the seller is subject to seller limits. (For more information, see "Seller Limits" on page 174.) The returned information is the remaining quantity of items that may be listed and the remaining listing value. If these limits are exceeded, the `AddItem` attempt will fail. So you should invoke `GetMyeBaySelling` before calling `AddItem` or `AddItems` to make sure the limits will not be exceeded in the call..

The `AddItem` call lists one item. For information about `AddItem`, see the [AddItem](#) page in the Call Reference.

The `AddItems` call lists up to five items per request. For multiple items, you can use `AddItem` repeatedly (specifying a different item each time), but using `AddItems` provides better performance and reduces the number of required calls. Refer to the [AddItems](#) page in the Call Reference for more information.

An application can test a listing with the `VerifyAddItem` call. This call uses the same inputs, and returns the same values, as the `AddItem`, except that `VerifyAddItem` does not actually list the item. An application could use `VerifyAddItem` to test an item definition, and give the seller a preview of the estimated listing fees, before actually listing the item. Then the application could use the same `ItemType` object again, unchanged, with `AddItem`.

Sellers who engage in cross-border trade on sites that require a recoupment agreement must agree to the recoupment terms before using the [AddItem](#) or the [VerifyAddItem](#). This agreement allows eBay to reimburse a buyer during a dispute and then recoup the cost from the seller.

Check for a Message node in the [VerifyAddItem](#) response because it may indicate that an [AddItem](#) attempt will be unsuccessful; see the following Knowledge Base article:

VerifyAddItem Returns Success with No Errors, but AddItem Returns Failure with Errors
https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=912

To modify a listing after it has been submitted, use [Reviseltem](#) or [AddToltemDescription](#).

To relist an unsold item, use [RelistItem](#). You can use [VerifyRelistItem](#) to test [RelistItem](#).

See “[Managing Item Listings](#)” on page 375 for additional information about these calls.

This section contains the following subsections:

- “[Required Information for eBay Listings](#)” on page 177
- “[Optional Information for eBay Listings](#)” on page 181
- “[Considering Best Match When Listing Items](#)” on page 184
- “[Considering Best Match When Revising and Relisting Items](#)” on page 184

Required Information for eBay Listings

To define a new item, create an [ItemType](#) object with the new item’s definition, and assign values to the [ItemType](#) object’s properties. There are a number of item properties that can be set, some required and some optional. All define how the item is displayed, how a buyer needs to go about purchasing the item, and what actions need to occur in the checkout process. See the schema documentation for [ItemType](#) for a complete list of its properties.

This section summarizes some of the information that most eBay listings require. For additional required fields, refer to the [eBay Trading API Call Reference](#).

Title and Description

The main descriptor is the item’s title. This value is seen on the most pages and, while brief, should be composed with the intent of drawing the attention of potential buyers interested in that type of item. A title is required for item listings.

You can also optionally provide a subtitle for most items, the text of which is displayed on the View Item page and is used for item searches. Compose a description for the item that is relatively short, but that tells a user such things as what the item is, the condition of the item, any special instructions for a buyer, and anything else that would help make the item appealing to the intended buyer. Starting in October 2012, it is recommended that you use the [Item.ConditionDescription](#) field to describe used items, as opposed to describing the condition of that item in the [Item.Description](#) field. For more information about describing the condition of a used item, see “[Describing Details About the Condition of Items That Are Not New](#)” on page 221.

The View Item page displays the title and description within an iFrame. The following are some steps you can take to make sure your content appears as expected inside an iFrame.

Content Dimensions

- All HTML content should have relative widths and heights.
- Images and other fixed-size objects should not exceed 700 pixels in width.

File Linking

- HTML and Flash links and form actions should target a _blank window and use full paths including the `http://` prefix.
- Externally hosted images and other content references should point to valid existing content and use full paths, including the `http://` prefix. In other words, there should be no broken links.
- Externally hosted images and other content should have quick load speeds.

Content Validity

- HTML content should be well formed and preferably should comply with XHTML 1.0 standards.
- Script and other rich content should not depend on any elements in the eBay View Item page.
- Scripts must properly execute without any runtime warnings or errors.

Site and Categories

The seller must designate the eBay site on which the item will be listed (see “[Standard Input Data](#)” on page 92). The seller must also specify a category on the selected site. The seller has the option to specify a second category for the same item on the same site. If a second category is selected, the item is listed simultaneously in the two categories. For more information on categories, see “[Categories \(Introduction\)](#)” on page 138. For information on retrieving category details, see “[Retrieving the Category Hierarchy for a Site](#)” on page 158.

If you define an item and a problem occurs with the category ID you specified, see “[Mapping Old Category IDs to Current IDs](#)” on page 163 for information about how to keep category data for individual items synchronized with eBay’s category data.

Store Categories

If the seller is an eBay Store owner and wants to list the item in their storefront, the seller can optionally specify up to two custom categories defined for that storefront by using `Storefront.StoreCategoryID` and `Storefront.Category2ID` in addition to specifying the primary and secondary eBay categories.

Free Added Category

eBay may choose at times to promote a listing with a free second category. eBay does this to help increase the visibility of the item to buyers who may be browsing in a category different than the one specified for the listing. When eBay adds a category, sellers are not charged any List in Two Categories fees for the listing. eBay may choose to add a second category to listings even on sites that normally do not support the ability to list in two categories.

The `AddItem` family of calls does not return any information about free added categories. However, `GetItem` and `GetSellerList` return information about the added category in a `FreeAddedCategory` (and `SecondaryCategory` is not returned).

If a seller submits a listing with two categories and it qualifies for a free second category, the fees returned from `AddItem` will reflect charges for listing in two categories, but the seller will receive a refund of any List in Two Categories fees for that listing. In this situation, `GetItem` and related calls return both categories in the `PrimaryCategory` and `SecondaryCategory` nodes as usual and `FreeAddedCategory` is not returned.

Price

A new listing must have a starting price. For an auction listing, the starting price is the price at which

Note: Certain sites prevent a seller from listing an item if the item's starting price is less than the site's minimum listing amount for the selected listing format. The minimum listing amount (specified in the site's supported currency) applies to the following sites: CanadaFrench (CAFR), Canada (CA), India (IN), eBay Motors, Netherlands (NL), Philippines (PH), Singapore (SG), Spain (ES), United Kingdom (UK) and US. For complete and updated information on the minimum listing amounts for various listing formats on these sites, along with dates when these changes go into effect, see http://pages.ebay.com/help/sell/starting_price.html and <http://pages.ebay.com/help/sell/fees.html> for the US and eBay Motors sites. For other sites, use the site abbreviations above. For example, the following would be valid URLs: http://pages.ebay.co.in/help/sell/starting_price.html or <http://pages.ebay.de/help/sell/fees.html>.

the progressive bidding starts. For a fixed-price listing, the starting price is the static price at which a buyer can purchase the item. Sellers can also add the Buy It Now feature to an auction listing, or assign a reserve price to the listing. If a reserve price is set, the item will not sell unless the bid of the winning bidder is higher than this price.

Payment Methods

Every listing must contain at least one payment method, and all specified payment methods should be supported by the site and by the category. For more information, see “[Specifying a Payment Method](#)” on page 184.

Quantity

A fixed-price listing may offer one item, or a multiple quantity of items. A fixed-price listing ends when the last item has been sold (if the listing duration has not yet passed and ended the listing).

Listing Type

Specify the listing type to indicate if the item will sell for a static, fixed-price (fixed-price listing), or sell using a bidding process (auction listing). A listing type is required for a new item listing. If not specified in an API call, the listing type will default to an auction listing. Some listing types have other seller qualifications. For information on all listing types, see “[Listing Types \(Formats\)](#)” on page 136.

Duration

A listing’s duration is the time (expressed in days) that the listing will be active on the eBay site. For auction listings (see “[Listing Types \(Formats\)](#)” on page 136), the listing ends and the winning bidder is determined when this time has passed. For fixed-price listings, the listing ends when this time has passed (though the listing could end earlier if all of the listing’s items are sold or the seller ends the listing early).

Item Location

The seller may specify the location of the item and the country of that location. An item’s location is a free-form text value, such as “San Jose, CA.” The country is indicated using the ISO 3166 two-letter abbreviation for the country, such as “us” for the United States. If specify a postal code, but do not specify a location, then the location is given a default value derived from the postal code.

By specifying the postal code where the item is located, the seller opts in to the ability for others to search for the item based on distance from a specified zip or postal code. For a list of valid postal code formats for countries that support search by proximity, see “[Postal and Zip Codes \(PostalCode\)](#)” on page 603.

The country is defined using the eBay 2-letter country code set.

Use the [GeteBayDetails](#) call to see the list of currently supported codes, and the English names associated with each code (e.g., KY="Cayman Islands").

Most of the codes that eBay uses conform to the ISO 3166 standard, but some of the codes in the ISO 3166 standard are not used by eBay. Plus, there are some non-ISO codes in the eBay list. (Additional codes appear at the end of this code list and are noted as non-ISO.)

Shipping

At listing time, the seller selects between using flat-rate shipping or calculated shipping. For flat-rate shipping, the seller specifies all shipping costs upfront based on the buyer's location and the shipping service selected by the buyer. In calculated shipping, eBay calculates shipping costs based on the shipping specifications (package dimensions and the weight of the item, or items if the seller is shipping multiple items to the buyer in the same shipment).

When listing in certain categories, a seller is required to provide at least one domestic shipping service. [GetCategoryFeatures](#) call can be used to find out, if a category needs shipping details provided or not. All categories with [Category.ShippingTermsRequired](#) set to `true` will require at least one domestic shipping service provided.

For more on working with shipping services and costs, see “[Shipping](#)” on page 312.

Handling (Dispatch) Time

For most items, the seller needs to specify a handling time. This indicates how long it will take for the seller to ship the item to a buyer after receiving cleared payment. This does not include the carrier's shipping time (in transit).

Top-rated sellers must offer a 1-day handling time (or 0-day handling time once the Same Day Handling feature becomes available) for a listing in order for that listing to be eligible as a Top-rated listing. The other requirement for a Top-rated listing is for the seller to accept returns for the item. Only sellers that qualify as Top-rated sellers can have Top-rated listings.

Shipping Regions

The seller can also indicate where, by geographical region, the item may be shipped. This is done by specifying one shipping option, indicating: the seller will only ship to users on the same site, the seller will ship to buyers worldwide, the seller will ship to buyers on the seller's site and specified regions, or the seller will not ship the item at all. The default is to ship to buyers on the same site. If the seller indicates he will ship to the same site plus additional regions, one or more shipping regions must be specified to indicate what those additional regions are.

Return Policy

Sellers who provide a clear return policy are likely to sell a higher percentage of their listed items than sellers who do not have a clear return policy.

Sellers are required to specify a return policy for auction and fixed-price listings in most categories. Sellers can choose not to accept returns, but they must still specify this policy clearly. See “[Offering a Clear Return Policy](#)” on page 223.

Top-rated sellers must accept returns for a listing in order for that listing to be eligible as a Top-rated listing. Only sellers that qualify as Top-rated sellers can have Top-rated listings.

Other Shipping Rules

Some sites, such as India, have specific shipping rules. For instance, an international listing fee applies when the seller lists items on the India site that are available for shipping outside of India.

Optional Information for eBay Listings

This section summarizes some of the additional information that can be included in a listing. For additional fields, refer to the [eBay Trading API Call Reference](#).

Pictures

As of October 1, 2012, all listings must have at least one picture of the item. Up to 12 pictures may be included with each listing. Pictures are added to a listing free of charge for all categories except ebay Motors vehicle listings. Item pictures are not stored with the item. Rather, only the URLs pointing to pictures are stored with the item's data in the database at eBay. The actual picture file may be stored either at eBay using the fee-based eBay Picture Services (EPS) or at a location of the sellers choice (such as an image hosting vendor or the seller's own Web site). The URLs for up to twelve images may be specified for an item if they are hosted on EPS. Special image effects (like image super-sizing) can be specified for EPS-hosted images. If image hosting other than EPS is used, then only one image may be associated with the item (one URL specified for the item definition). For specific information on associating images with an item listing, see “[Associating Pictures with an Item](#)” on page 209.

Fast Shipping and Handling

Many buyers want or need to receive an item as soon as possible after purchase (for example, last-minute shopping for the holidays). Such buyers benefit from knowing if purchasing an item from one seller would get the item into the buyer's hands faster than if purchased from another seller.

Sellers can specify prompt domestic handling and expedited shipping for items so buyers can assess whether items can be delivered within an acceptable time range. Some eBay sites support the Get It Fast feature. For an item to qualify for Get It Fast, the seller must offer at least one domestic, expedited shipping service and specify a domestic handling time of 1 business day (or 0 days once the Same Day Handling feature becomes available. For more information about Get It Fast, see “[Enabling Get It Fast](#)” on page 233.

Listing Enhancements

Listings on eBay may be enhanced to make them stand out from other listings, such as on listing browse and search pages where there are multiple items from different sellers displayed on the same page. One enhancement is to show the item's title on these pages in a bold typeface. Another listing enhancement is showing the whole item row on a listing page in a different (highlight) color. Listing enhancements are optional and all incur additional listing fees.

Skype

For listings in some categories on some sites, you can add Skype contact information. For more information, see “[Enabling Communication Through Skype](#)” on page 405.

Listing Designer

The description for an item may optionally be visually enhanced using Listing Designer templates and layouts. This makes the description as it appears on the View Item page more appealing to prospective buyers. Themes are visual accoutrements used in the item description. Themes are seasonal, based on a particular holiday, and based on personal events (like a wedding anniversary or a baby’s birth). Layouts control where an item’s image(s) appear in the description relative to the description’s text. The seller specifies the ID for one theme and/or the ID for one layout. (Retrieving Listing Designer theme and layout IDs is not implemented in this release of the API. See “[GetDescriptionTemplates](#)” on page 827 for more information.)

Donate a Percentage of the Sale

Specify whether or not to donate a percentage of the final sale price to a qualified nonprofit through the eBay Giving Works provider. These listings appear in the eBay search engine, in the eBay Giving Works search engine, and on a dedicated page for the nonprofit that appears on the eBay Giving Works provider Web site. Listings also contain an eBay Giving Works icon and specific information about the donation commitment and benefiting nonprofit. For more information on eBay Giving Works listings, see “[Identifying Listings that Benefit Nonprofits](#)” on page 234.

Item Specifics

To increase the chances that your listing description will include words that buyers will search on, include Item Specifics, which are well-known features that distinguish a particular type of item according to its category. For example, in a PDA category, you could specify that the “Screen” is “Color”; whereas in a book category you could specify a “Publication Year” with a value like “1999” (and screen color would have no meaning). In most categories, the use of Item Specifics is optional. A few categories (e.g., US Real Estate) require a seller to fill in values for Item Specifics that are marked as required. Some categories do not support Item Specifics at all.

See “[Working with Custom Item Specifics](#)” on page 481 for information about the data you need in order to include Item Specifics in a listing.

If you are new to the API, it may be easiest to first develop and test the listing functionality of your application without using Item Specifics, and then add support for Item Specifics once you have successfully created a number of listings.

Catalog Product Details

To make it even easier for buyers to find your listing, include product details from a catalog in the listing. With this option, you search an informational catalog for a stock product that is similar to your item, and eBay uses that product’s stock content to fill in a portion of your listing’s Item Specifics and to add other information to supplement your listing description. The eBay catalog contains descriptive information only (no actual physical products). This option is available in certain categories only.

To indicate that you want to include this kind of data in your listing, you need to specify an eBay product ID (EPID) or a UPC, ISBN, EAN, or MPN. See “[Pre-filling Item Specifics with Product](#)

[Details](#) on page 484 for more information about product details and the data you need in order to use stock information in a listing.

If you are new to the API, it may be easiest to first develop and test the listing functionality of your application without using product details, and then add the product details once you have successfully created a number of listings the standard way.

Combined Payment

If multiple, unpaid order line items exist between the same buyer and seller, that buyer or seller have the option of combining those multiple order line items into one order. The benefits of a Combined Payment order is that only one payment is made for all order line items, and the seller is often in the position to offer the buyer discounted shipping due to the fact that it is possible to ship multiple items in one shipment, and this saves the seller on shipping costs. Shipping discount rules for flat-rate and calculated shipping are set and managed through Combined Payments and Shipping Discounts preferences in My eBay. Optionally, shipping discount rules can be set and managed through the [SetShippingDiscountProfiles](#) API call. The shipping discounts are set in the [CalculatedHandlingDiscount](#), [CalculatedShippingDiscount](#), and [FlatShippingDiscount](#) containers and will apply towards eBay orders if the Combined Payment option is enabled.

As a seller, you control whether or not you allow buyers (or yourself) to create Combined Payment orders. The Combined Payment options are also found in the Combined Payments and Shipping Discounts preferences in My eBay. In the API, the Combined Payment options are set and managed with the [SetUserPreferences](#) call. The Combined Payment option is turned on with the [CombinedPaymentOption](#) flag in [SetUserPreferences](#), and the maximum amount of days (the default is 30 days) in which unpaid order line items can be combined into a Combined Payment order is set with the [CombinedPaymentOption](#) field in [SetUserPreferences](#).

Gift Services

The seller may opt to have special Gift Services features used for the item's shipping to the buyer: gift wrapping and a gift card, express shipping, and shipment to the gift recipient instead of the buyer. If Gift Services are used, the gift icon must also be specified.

Other Information

There are also a variety of other properties for an item that can be set. This includes adding a hit counter to the item's View Item page. A seller may also designate a competitive-bidding or fixed-price item as a private listing. When private listings are bid on or bought, the IDs of the bidders and the buyer are hidden for privacy reasons. A seller may also supply data that is applicable or meaningful only to the listing application. This application data is stored with the item data in the eBay database, but is never used by eBay for any purpose. It is there solely for the application's use.

[AddItem](#) returns the item ID (the unique identifier for the new listing), along with the fees the seller can expect to pay for the listing.

Before using [AddItem](#), you can also pass the item data in [VerifyAddItem](#) to validate the item data and determine the estimated fees.

See "[Fees Resulting from Listing an Item](#)" on page 189 for more information.

As [GetItem](#) doesn't return an item's listing fees, you should save the fees returned from [AddItem](#) if you want to keep a record of the expected fees.

Considering Best Match When Listing Items

When you list an item, you can consider how to increase its placement, or rank, in search results. When users search for items on eBay, the default way items are sorted is Best Match.

For information about Best Match, including the definition of the “performance score,” see “[Considering Best Match When Revising and Relisting Items](#)” on page 184 and the following locations:

BestMatchItemDetails API

<https://www.x.com/developers/ebay/products/best-match-api>

Considering Best Match When Revising and Relisting Items

When you consider revising and relisting items, consider the items’ performance score (i.e., the ratio of sales to impressions). An item’s performance score affects the item’s rank in search results. For more information about the performance score, see the following:

BestMatchItemDetails User Guide

<http://developer.ebay.com/DevZone/best-match-item-details/Concepts/BestMatchItemDetailsAPIGuide.html>

Revising an item doesn’t affect its performance score, in most cases. Similarly, relisting an item will carry over an item’s performance score, regardless of changes made, in most cases. However, whether you are revising or relisting, either of the following changes will reset an item’s performance score:

- Category updates (changing, adding, or removing) for either the primary or secondary category will trigger a reset
- Item condition removal

Note: When relisting an item, the item performance score carries over for the first new item only. That is, if an item is relisted multiple times, only the first new item inherits the performance score. When it is time to relist the new item, its performance score carries over for the first new item only, as well.

For more information about Best Match, “see “[Considering Best Match When Listing Items](#)” on page 184 and the following location:

Specifying a Payment Method

This section contains general information about specifying payment methods in a listing, along with information about specifying ‘PayPal’ as a payment method. (For information about the [PaisaPay](#) payment method, see “[Payment Method Differences \(PaymentMethod\)](#)” on page 601.)

Each listing must contain at least one valid (for the site and category) payment method. You can specify a payment method through the [AddItem](#) call. In the example below, ‘PayPal’ is specified as a

payment method. You can actually specify numerous, valid payment methods through multiple **PaymentMethods** fields.

Example 9-1 Specifying a Payment Method

```
...  
<Item>  
...  
<PaymentMethods>PayPal</PaymentMethods>  
...  
</Item>  
...
```

Note that if a seller specifies PayPal as one of the payment methods, the seller must also supply his or her PayPal email address through the [Item.PayPalEmailAddress](#) field. This value is used to identify the seller as the recipient of the buyer's payment if the buyer pays for the item using PayPal. The seller may optionally create an immediate payment item (see ["Requiring Immediate Payment"](#) on page 187). In an immediate payment listing, the sole payment method must be PayPal.

Also note that the [GetCategories](#) call can be used to determine whether the [AutoPay](#) option is valid for listing on a site and in specific categories. For more information, see ["GetCategories"](#) on page 805.

When you use [AddItem](#) to specify a payment method that is invalid for the target site, the invalid payment method is ignored and the item is listed (although a warning about the invalid payment method(s) is returned). If all specified payment methods are invalid for the target site, the default payment method for the site is used. If necessary, the default payment method for a site can be discovered by looking at the [SiteDefaults.PaymentMethod](#) value in the [GetCategoryFeatures](#) response.

After you list an item, you can use the values in the [PaymentAllowedSite](#) field returned by [GetItem](#) to confirm the sites on which the item can be purchased, based on the payment methods you offered.

When you revise or relist an item, rules apply for changing the existing payment methods. See ["Changing Payment Methods in a Relisted Item"](#) on page 378 and ["Changing Payment Methods When Revising an Item"](#) on page 384.

Determining the Payment Methods Allowed for a Category

Use [GetCategoryFeatures](#) to determine the payment methods that are allowed for a category on a site. For example, the response data of [GetCategoryFeatures](#) will show that on the US site, most categories only allow electronic payments. Also use [GetCategoryFeatures](#) to determine the default payment method for a site by looking at the [SiteDefaults.PaymentMethod](#) value. Do not use [GeteBayDetails](#) to discover the valid payment methods for a site.

In the following **GetCategoryFeatures** request, **GetCategoryFeatures** is used to determine the payment methods applicable to the Children's Books category:

Example 9-2 Using GetCategoryFeatures to Determine Applicable Payment Method

```
<?xml version="1.0" encoding="utf-8"?>
<GetCategoryFeaturesRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <RequesterCredentials>
        <eBayAuthToken>ABC...123</eBayAuthToken>
    </RequesterCredentials>
    <Version>585</Version>
    <ViewAllNodes>true</ViewAllNodes>
    <DetailLevel>ReturnAll</DetailLevel>
    <CategoryID>279</CategoryID>
    <FeatureID>PaymentMethods</FeatureID>
</GetCategoryFeaturesRequest>
```

You can use **GetCategoryFeatures** to determine if listings on a site must include PayPal as a payment method. Most categories on the U.S., Canada, and Australia sites require sellers who have registered after January 17, 2007 to offer a safe payment method, such as PayPal, PaisaPay (India only), or one of the credit cards specified in **Item.PaymentMethods**. The **SafePaymentRequired** boolean field will be returned in **GetCategoryFeatures** if a safe payment method is required.

For additional information about payment methods, see the following location:

Accepted Payments Policy

<http://pages.ebay.com/help/policies/accepted-payments-policy.html>

Using Integrated Merchant Credit Card (IMCC)

On the US site, in **AddItem** and related calls, you can use the **IntegratedMerchantCreditCard** payment method if the seller of the item has a payment gateway account. The payment gateway account can be set up through options available to the seller in My eBay. For information about credit and debit card processing (through merchant accounts), see the following location:

Expanded credit/debit card processing through merchant accounts

<http://pages.ebay.com/sellerinformation/news/imcc2010.html>

If a seller uses the **IntegratedMerchantCreditCard** payment method in **AddItem** for a listing, it has the benefit of replacing the use of specific credit-card payment methods, such as **VisaMC**. If a seller uses the **IntegratedMerchantCreditCard** payment method for a listing, the listing displays (to potential buyers) the credit cards that the seller specified in the seller's preferences for their payment gateway account (in My eBay). Additionally, a buyer's credit card payment is integrated into eBay checkout.

In **GetItem** and related calls, **Item.IntegratedMerchantCreditCardEnabled** indicates if, on checkout, an online credit-card payment is processed through a payment gateway account.

If a payment-gateway-registered seller is using **RelistItem** or **RevisedItem** for a listing that was submitted with specific credit-card payment methods, the item will automatically be relisted/revised with the **IntegratedMerchantCreditCard** payment method. If a seller is not payment-

gateway-registered, the **IntegratedMerchantCreditCard** payment method will get reset and the item will get listed with other specified payment methods (if any) other than **IntegratedMerchantCreditCard**. If no other payment methods were specified in the original listing, the listing will be blocked.

The **User.SellerInfo.IntegratedMerchantCreditCardInfo.SupportedSite** field, returned by [GetUser](#), indicates the sites on which a seller has a payment gateway account.

Site Visibility

The payment method differences among eBay sites can affect the visibility of an item on different sites. For example, on the Australia (AU) site, only **PayPal** and **CashOnPickup** can be specified in the **PaymentMethods** field. If a non-AU seller specifies shipment to Australia in an item listing, but does not specify a valid payment method for the AU site, then the item will be unavailable in the search results of the AU site. If that listing contains PayPal but also contains payment methods that do not apply to the AU site, then only the PayPal payment method is shown on the AU site.

Requiring Immediate Payment

For a Buy It Now auction listing or a fixed-price listing, qualified sellers have the option of requiring a buyer to pay immediately after committing to purchase an item. This is called *immediate payment*.

Listings in some categories (e.g., Tickets) might lose their value if they are not sold by a particular date. For listings that contain time-sensitive (or *perishable*) items, eBay gives sellers the ability to specify that if the buyer pays the Buy It Now price or Fixed Price for the item, the payment must be made immediately through PayPal. Normally, eBay ends a listing and creates an order line item when a buyer agrees to purchase an item. If the seller chooses to require immediate payment, eBay ends the listing (or decrements the quantity in a multi-item listing) and creates an order line item after the payment has been processed. See the immediate payment online help on the eBay site for additional information about choosing to require immediate payment.

Requiring Immediate Payment: Overview
<http://pages.ebay.com/help/pay/require-immediate-payment.html>

Requirements for Using Immediate Payment with AddItem

A seller can choose to require immediate payment for a Fixed-Price listing, or for an auction listing enabled with a Buy It Now option. If a Buy It Now auction listing ends as an auction (that is, if the Buy It Now option is removed due to bids being placed on the listing), the immediate payment requirement does not apply.

If a seller wants to require immediate payment, several conditions apply:

- The category must support immediate payment. The **AutoPayEnabled** flag must be **true** for the category in the **GetCategories** response.
- The seller must specify PayPal as the only method of payment (only one **PaymentMethods** field set to **PayPal**), and they cannot offer escrow.
- The seller must be registered as an eBay user on a site that supports immediate payment (see “[Payment Method Differences \(PaymentMethod\)](#)” on page 601) and the item must be listed in the seller’s geographic location (that is, **Item.Site** must be consistent with the seller’s registration address). Items offered with immediate payment can be shipped abroad as long as shipping costs are specified for the shipping location.
- The seller must have a Business or Premier PayPal account and be in good standing with eBay. If the seller is in good standing, **GetUser** returns a value of true in the **User.SellerInfo.GoodStanding** field.
- The price passed in the item’s **BuyItNowPrice** field (for auction listings) or **StartPrice** (for Fixed Price listings) cannot be higher than \$10,000 USD.
- The seller must specify shipping costs and cannot offer a shipping discount for multiple purchases (see “[Combined Payment](#)” on page 183).
- The **Item.PayPalEmailAddress** field is required, and it must contain a valid payment email address. The corresponding PayPal account cannot be in a High Restricted or Locked state.
- The seller must specify all related costs to the buyer, since the buyer will not be able to use the Buyer Request Total feature in an immediate payment listing; these costs include flat-rate shipping costs for each domestic and international shipping service offered, package handling costs, and any shipping surcharges.
- The seller must include and set the **PromotionalShippingDiscount** field to true if a promotional shipping discount is being applied to the listing.
- The seller must include the **ShippingDiscountProfileID** and reference a valid flat or calculated Shipping Discount Profile ID if a flat or calculated shipping rule is being applied to the listing.

To indicate that the seller has chosen to require immediate payment, set the item’s **AutoPay** field to true when using **AddItem** (or **Reviseltem**).

See “[Requiring Immediate Payment of a Vehicle Deposit](#)” on page 569 for additional rules that apply when the seller requires Immediate Payment of a US eBay Motors vehicle deposit.

Note: German (DE) and Austrian (AT) sellers who are subject to the new eBay payment process are restricted from creating an immediate payment listing.

Testing the Immediate Payment Feature

When testing this feature in the Sandbox, make sure **AddItem** has **Item.AutoPay** set to “true” and **Item.PaymentMethods** set to “PayPal”. To test your PayPal Sandbox email address, include it in **Item.PayPalEmailAddress**.

Changing the Immediate Payment Requirement for an Item

When using [RevisedItem](#) or [RelistItem](#), the default value of the `Item.AutoPay` flag is the value the seller used for the original listing.

To determine whether or not a seller requested immediate payment for a particular listing, use [GetItem](#). The call returns the `Item.AutoPay` flag with a value of true if immediate payment was specified.

Note: The value of the `AutoPay` flag indicates the seller's stated preference only. It does not indicate whether the listing is still a candidate for purchase via immediate payment. For example, if a listing receives bids and no longer qualifies for immediate payment, the value of the `AutoPay` flag does not change.

If immediate payment is specified and the seller attempts to list the item in a category that does not support this feature, eBay will return an error. The seller should choose an appropriate category or remove the immediate payment requirement. (Items in certain categories may be subject to additional standard rules with respect to category changes.)

Note: If a listing does not specify immediate payment, a seller can add a requirement for immediate payment when revising the listing as long as all the conditions specified in "Requirements for Using Immediate Payment with AddItem" on page 187 are met. The usual rules for revising a listing must also be met (that is, the listing has no bids or purchases, and the listing does not end within 12 hours).

Similarly, a seller can remove a request for immediate payment from an existing listing if the usual rules for revising an item are met. In this case, set `AutoPay` to false. The seller can then choose additional payment methods in addition to `Item.PaymentMethods` value of `PayPal` for the item. However, the seller cannot modify the item's shipping costs.

If immediate payment was originally specified for a Buy It Now listing and the Buy It Now option no longer applies (e.g., a bid was placed), eBay automatically removes the immediate payment requirement on behalf of the seller. However, eBay does not change the value of `Item.AutoPay`, so it will still be returned as True by the [GetItem](#).

Fees Resulting from Listing an Item

This section details the costs incurred when you list an item on eBay. Topics include:

- "[Fees Associated with Listing Features](#)" on page 189
- "[Promotional Discounts](#)" on page 193

Fees Associated with Listing Features

When you list an item on eBay, you can enhance your listing by incorporating various *listing features* with your item listing. For example, you can bold your title, add a border to the listing, or add pictures and custom designs to draw attention to your item. Each listing feature has an associated cost, which is returned in a `Fee` container in the responses to the [AddItem](#) family of calls.

Responses from the `AddItem` family of calls include a `Fee` container for each eBay listing feature. As shown in the following example, each `Fee` container has both a `Name` and a `Fee` element, representing the name and cost of an eBay listing feature, respectively:

```
<Fee>
  <Name>BoldFee</Name>
  <Fee currencyID="USD">2.0</Fee>
</Fee>
```

Note that eBay returns `Fee` containers for all available listing features, even if their cost is zero.

Note: `VerifyAddItem` may return details about fees and discounts that applied at the time you ran `VerifyAddItem`. This is no guarantee that those fees and discounts will apply at the time you actually list the item.

The table below summarizes the types of fees that can be charged when you list an item to a US or international eBay.com site (including eBay Motors). The cost of all fees incurred when you list an item are summed and returned as the `ListingFee`.

The total cost of all *listing features* is found in the `Fees` container whose `Name` is `ListingFee`. This does not reflect the full cost of listing and selling an item on eBay, for the Final Value Fee cannot be calculated by eBay until the listing has ended, when a final sale price is known. Total cost is then the sum of the Final Value Fee and the `Fee` corresponding to `ListingFee`. See “[Final Value Fees and Credits](#)” on page 425 for more information.

A seller’s account can also include other listing fees and credits, such as promotion credits, unpaid item dispute credits, and `ReserveFee` refunds. To retrieve this additional fee information, use `GetAccount`.

If you need more details about the fees for a particular eBay site, we have provided links to the applicable Fee help pages in “[Fees per Site](#)” on page 593.

Fees Resulting from Listing an Item

Fee Name	Related AddItem Input Fields	Description
AuctionLengthFee	<code>ListingDuration</code>	Fee for 10-day listings. In the eBay US Web site UI, this is the “10-Day Duration” fee.
BasicUpgradePackBundleFee	<code>ListingEnhancement</code> [<code>BasicUpgradePackBundle</code>]	No longer applicable to any site. Formerly, a discounted fee for items listed with the following “bundle” of listing upgrades: Gallery and Subtitle. This upgrade formerly was available on the eBay Australia site only. This upgrade on that site has been replaced by the ValuePackBundleFee discount.
BoldFee	<code>ListingEnhancement</code> [<code>BoldTitle</code>]	Fee to boldface the listing title in the eBay Web site UI’s search results pages and category listings pages.
BorderFee	<code>ListingEnhancement</code> [<code>Border</code>]	Fee to display a border around an item in the eBay Web site UI’s search results pages and category listings pages.
BuyItNowFee	<code>BuyItNowPrice</code>	Fee to add the Buy It Now option to an item.

Fees Resulting from Listing an Item (Continued)

Fee Name	Related AddItem Input Fields	Description
CategoryFeaturedFee	ListingEnhancement [Featured]	Fee to display the item prominently in the Featured Items section in category listings in the eBay Web site UI, and at the top of the Web site search results page on which it appears (not necessarily the first page of search results). It will also display in the regular, non-featured item list.
CrossBorderTradeGBFee	CrossBorderTrade [UK]	Fee for the item to appear in the respective site's default search results.
CrossBorderTradeNorthAmericaFee	CrossBorderTrade [North America]	Fee for the item to appear in the respective sites' default search results.
FeaturedFee	ListingEnhancement [HomePageFeatured]	Fee for the chance to rotate the item into a special display on eBay's Home page. In the eBay Web site UI, this is the "Home Page Featured" fee. The item is very likely to show up on the Home page, although eBay does not guarantee that the item will be highlighted in this way.
FeaturedGalleryFee	PictureDetails.GalleryType [GalleryFeatured]	Fee to have the item appear in the special Featured section above the general picture gallery in the eBay Web site UI (in search results and category listings). In the eBay Web site UI, this is the "Gallery Featured" fee.
FixedPriceDurationFee	ListingDuration	Fee for listing a fixed price item with certain durations.
GalleryFee	PictureDetails.GalleryType [Featured]	Fee to have the item's picture included in the eBay Web site UI's search results and category listings pages.
GalleryPlusFee	PictureDetails.GalleryType [Plus]	Fee to display a larger picture of the item in the eBay Web site UI search results.
GiftIconFee	GiftIcon and GiftServices	Fee for displaying a gift icon next to the listing in the eBay Web site UI's search results and category listings pages. Some sites may also charge a fee if you include gift services search capabilities in the listing. In the eBay Web site UI, this is the "Gift Services" fee.
HighLightFee	ListingEnhancement [Highlight]	Fee to have the item's listing appear highlighted in the eBay Web site UI's search results and category listings pages.
InsertionFee	(no specific fields)	When you list an item, you're charged an Insertion Fee for the listing. InsertionFee is based on the starting price (or reserve price) and the category of your item.

Fees Resulting from Listing an Item (Continued)

Fee Name	Related AddItem Input Fields	Description
InternationalInsertionFee	InternationalShippingServiceOption	For sites that don't normally charge insertion fees, a fee for offering shipping to countries other than the country of the listing site.
ListingDesignerFee	ListingDesigner	Fee charged for the optional use of a Listing Designer layout or theme template.
ListingFee	(no specific fields)	Total fee for listing the item. Includes basic fee (InsertionFee) plus any listing upgrades (e.g., BoldFee). Note that the final cost of your item listing can include a Final Value Fee, which cannot be determined until the completion of the listing. See also " Value-Added Tax (VAT) " on page 586. For ReviselItem , this is the change in fees due to the revision actions.
MotorsGermanySearchFee	MotorsGermanySearchable	Fee to allow the item to be included in search results on the mobile.de site. Only charged if MotorsGermanySearchable is set to true for the listing and the category is mobile.de search-enabled on eBay Germany. See " Featuring Vehicles on the Mobile.de Site " on page 239 for additional information and a list of valid categories.
PhotoDisplayFee	PictureDetails.PhotoDisplay	Fee for special display of an item's pictures (such as super-size). See " Photo Fees " on page 213.
PhotoFee	PictureDetails.PictureURL	Fee for including pictures in a listing. On some sites, one or more pictures may be free. Also, some categories (e.g., Motors vehicles) may support more free pictures than others. See " Photo Fees " on page 213 and " Fees per Site " on page 593.
PrivateListingFee	PrivateListing	Fee for adding a private listing or making a listing private. When you create a private listing, a buyer's user ID will not appear in the listing or in the listing's bid history. Sellers may want to use this option when they believe that potential bidders for a listing would not want their user IDs disclosed to others. This fee is only charged on the eBay Germany and Austria sites. (That is, the eBay US site does not charge this fee.)
ProPackBundleFee	ListingEnhancement [ProPackBundle]	Discounted fee for items that were listed with the following "bundle" of listing upgrades: Bold, Border, Highlight, and Featured Plus!. In the eBay Web site UI, this is the "Pro Pack for Motors" fee. This upgrade is only available to US and Canadian eBay motor vehicle sellers, except in the Parts and Accessories category.

Fees Resulting from Listing an Item (Continued)

Fee Name	Related AddItem Input Fields	Description
ProPackPlusBundleFee	<code>ListingEnhancement</code> <code>[ProPackPlusBundle]</code>	Discounted fee for items that were listed with the following “bundle” of listing upgrades: BoldTitle, Border, Highlight, Gallery Featured, and Featured Plus!. In the eBay Web site UI, this is the “Pro Pack” fee. Support for this feature varies by site and category. This upgrade is available on the US and Canada sites (but not for motor vehicles).
ReserveFee	<code>ReservePrice</code>	Fee for specifying a reserve price for an auction.
SchedulingFee	<code>ScheduleTime</code>	Fee for scheduling the item to be listed at a later date. In the eBay Web site UI, this is the “Scheduled Listings” fee.
SubtitleFee	<code>SubTitle</code>	Fee to add a subtitle to the item. Not applicable to US eBay Motors (Site ID 100) listings.
ValuePackBundleFee	<code>ListingEnhancement</code> <code>[ValuePackBundle]</code>	Discounted fee for items that were listed with the following “bundle” of upgrades: Gallery, Subtitle, and Listing Designer. Support for this feature varies by site and category. This upgrade is available on the US, Canada, and Australia sites (but not for motor vehicles).

Promotional Discounts

Listing features are enhancements that a seller can add to their item listings, such as (listing displayed in) Bold, setting a Reserve Price, and including a Subtitle. eBay charges a fee for each listing feature that the seller incorporates into their listing. At eBay’s discretion, the seller might receive a discount on one or more of the listing features related to listing an item. A seller might see on the eBay Web site, “Today only! Add highlight and bold to your listing for half the normal cost.”

If the seller receives a discount on a listing upgrade, the amount of money that the seller saved on the upgrade is returned in the `Fees.Fee.PromotionalDiscount` field of the `AddItem` response. Each `PromotionalDiscount` under a `Fees.Fee` node indicates the amount of the discount for that listing feature. This applies to the Add/Revise/Relist family of API calls.

For example, if there is a discount of \$1.50 for a \$2.00 Bold feature, you will see the following `Fee` container:

```
<Fee>
  <Name>BoldFee</Name>
  <Fee currencyID="USD">0.5</Fee>
  <PromotionalDiscount currencyID="USD">1.5</PromotionalDiscount>
</Fee>
```

When a discount has been applied, the **Fee** value represents the post-discounted cost for the particular listing feature. To determine what the fee would have been without the discount, add together the values of **Fee** and **PromotionalDiscount**.

The total cost of all listing features is returned in the **Fee** container whose **Name** value is **ListingFee**. If this container has a **PromotionalDiscount** field, the **PromotionalDiscount** value is the total of all the promotional discounts that were applied when the item was listed, and the **Fee** value is the total cost of all listing features minus the promotional discounts.

Chapter 10

Using PayPal

This chapter describes how PayPal works during the eBay checkout flow. In addition, the PayPal Purchase Protection program is explained and an overview is given on how to integrate PayPal into your eBay applications.

In addition to the PayPal sections in this chapter, refer to ["Holds on PayPal Payments"](#) on page 421 for more information on payments made through PayPal.

- ["PayPal Overview"](#) on page 195
- ["Offering PayPal Buyer Protection"](#) on page 196
- ["Integrating PayPal With Your eBay Applications"](#) on page 197
- ["Setting Up the PayPal Sandbox"](#) on page 199

PayPal Overview

PayPal is an online payment service. PayPal provides buyers with a safe and secure means to pay sellers for items they purchase. Although PayPal is a separate site, it is closely integrated with the eBay buying and selling process.

PayPal Home Page
<https://www.paypal.com/ebay>

The Trading API fully supports offering PayPal as a payment method. For information about specifying a payment method for a listing, see ["Specifying a Payment Method"](#) on page 184.

When building applications to take advantage of PayPal as a payment method, you must take into account a few special considerations when you list items. The most important is to include PayPal as one of the payment methods that potential buyers can use when purchasing your item. For more information on how to list items on eBay, see ["Listing an Item"](#) on page 176.

When listing an item, the seller can specify PayPal as one of the payment methods they will accept for the item. If, after purchasing the item, the buyer selects PayPal as the method for paying for the item, PayPal forwards the payment to the seller and manages parts of the checkout flow.

To use PayPal—for either sending payments as a buyer or receiving payments as a seller—the user must have a Personal, Premier, or Business PayPal account. For buyers, the account is the source of funds for orders. For sellers, the account is the destination for the payments received from buyers.

Sellers benefit from the PayPal service because it provides a safe, secure, and reliable means for getting the payment for an item from the buyer. The seller does not need to worry about whether the buyer would pay via check, credit card, or other means. This is transparent to the seller as PayPal simply transfers the buyer's payment to the seller's PayPal account.

Buyers benefit from PayPal because it is flexible and secure. A buyer may pay for items through PayPal by credit card, a bank account, or cash held in the user's name at PayPal. Additionally, PayPal offers PayPal Purchase Protection, as described in "Offering PayPal Buyer Protection" on page 196.

Offering PayPal Buyer Protection

Sellers cannot directly offer PayPal Purchase Protection in their listings. However, a seller can make certain decisions that will help ensure that the seller's listings are eligible for PayPal Purchase Protection.

For example, the seller should accept PayPal as a payment method so that the item can be considered for this form of protection. To accept PayPal payments, pass a value of **PayPal** in your array of **Item.PaymentMethods** fields.

Note for US Ticket Listings Only: New York and Illinois ticket resale laws require online platforms permitting sales of event tickets to offer full buyer protection. eBay offers full buyer protection on ticket sales to New York and Illinois events through PayPal. eBay therefore requires all sellers to accept PayPal payments for ticket listings when the venue state is New York or Illinois (regardless of where the buyer or seller is located).

For information about PayPal Purchase Protection, please see the PayPal site. The PayPal site contains information about the availability and requirements of PayPal Purchase Protection on US sites and on non-US sites. It also includes links to non-US PayPal sites.

PayPal Site

<http://www.paypal.com>

PayPal Purchase Protection Page

https://www.paypal.com/us/cgi-bin/webscr?cmd=xpt/Marketing_CommandDriven/general/PBInfo-outside

For information about buyer protection on the Australia site, see the following location:

PayPal Australian Buyer Protection Information

https://www.paypal.com/au/cgi-bin/webscr?cmd=p/gen/ua/policy_buyer_complaint-outside

On the Australia site, you can use **GetCategoryFeatures** to determine if purchase protection is supported in a category.

In general, PayPal Purchase Protection may apply when a buyer pays for a qualified item through PayPal. The **BuyerProtection** field of an item indicates whether the item is eligible for PayPal Buyer Protection. Eligibility of an item for PayPal Buyer Protection is determined after the item is listed (for example, after the seller has successfully used **AddItem** to list an item). Eligibility is based on the requirements described on the PayPal site.

You can use the following calls to retrieve information about items that were listed with buyer protection (these calls return the [BuyerProtection](#) field):

- [GetItem](#)
- [GetBidderList](#)
- [GetSellerList](#)
- [GetItemTransactions](#)
- [GetSellerTransactions](#)
- [GetOrderTransactions](#)
- [GetSellingManagerTemplates](#)

When an item is revised, the eligibility is re-evaluated if the seller made listing changes such as a new PayPal address or item category, but not if the seller merely updated the item description. Note that second chance offers (see “[Making Second Chance Offers for Items](#)” on page 426) are evaluated at the time the offer is made.

In rare cases, a combined payment (see “[Combined Payment](#)” on page 183) may include some items that are eligible for PayPal Buyer Protection and some items that are not. PayPal may extend coverage to ineligible items in such cases, even if the [BuyerProtection](#) field for the items indicates they are ineligible.

To ensure a better overall buying and selling environment, PayPal will in some cases place a hold on a buyer’s PayPal payment. To learn more about PayPal holds, and why they are placed, refer to “[Holds on PayPal Payments](#)” on page 421.

Integrating PayPal With Your eBay Applications

When you list an item on eBay (for example, by using [AddItem](#)), the response contains an eBay [ItemID](#), a value that uniquely identifies that item on the eBay site. When a buyer commits to buy an item, eBay generates a [TransactionID](#) value and an [OrderLineItemID](#) value to uniquely identify that order line item.

If PayPal is used as the payment method, you can use the corresponding [ItemID](#) and [TransactionID](#) values, or just the [OrderLineItemID](#) to retrieve the PayPal transaction ID. Using the PayPal transaction ID, you can follow the checkout flow through PayPal and identify the different stages of payment processing.

Note: PayPal has a Sandbox for testing purposes. You can test your PayPal integration in the Sandbox without performing real orders. For more information, see “[Setting Up the PayPal Sandbox](#)” on page 199

Obtaining the PayPal ExternalTransactionID

By passing the eBay [OrderLineItemID](#) value to the Trading API’s [GetItemTransactions](#) or [GetOrders](#) calls, you can retrieve PayPal’s transaction ID, which is returned in the [ExternalTransactionID](#) field of

the **ExternalTransaction** container. Using this value in PayPal API calls, you can gain the necessary information to track and complete the order.

For example, the PayPal **GetTransactionDetails** call is a good way to obtain details about a single PayPal transaction. In addition to payer and payment information (such as the gross payment, transaction date, fees, and so on), you can also receive information about the eBay order, such as the eBay Item ID (**L_NUMBER0**) and the eBay Transaction ID (**L_EBAYITEMTXNID0**).

The following Trading API calls return an **ExternalTransactionID** value:

- **GetItemTransactions**
- **GetSellerTransactions**
- **GetOrders**
- **GetOrderTransactions**

The PayPal APIs

You can use the **TransactionSearch** PayPal API call to search for PayPal transactions. Your search can be based on several criteria, including the time frame of a transaction, the eBay **ItemID**, the payer's name or email address, the receipt number, or the payment status.

For a full list of the available PayPal API calls and information on how to use them, see:

https://cms.paypal.com/us/cgi-bin/?cmd=_render-content&content_ID=developer/howto_api_reference

PayPal also offers a Developers Program, complete with a Sandbox testing environment and developer support.

PayPal Developer Central Page
<http://www.paypal.com/developer>

Using Instant Payment Notification

As an alternative to getting updates to payment status with frequent calls to PayPal's **TransactionSearch** call, you can use the PayPal *Instant Payment Notification* (IPN) service to receive updates from PayPal. Instant Payment Notification is a push technology where PayPal sends order updates whenever they occur. To receive Instant Payment Notifications, you need to set up a Web server.

If you use Instant Payment Notification, look for the eBay Transaction ID. Transaction IDs can be retrieved with **GetOrders** and other eBay order retrieval API calls.

The near real-time nature of IPN posts enables you to provide a high level of customer support. For example, if a buyer pays with an eCheck, the payment status may be **Pending** for several days while the check clears. In this case, you may choose to notify the buyer of the delay. Then, when you receive an IPN indicating payment has cleared, you can notify the buyer that the item will be shipped, and you can update your backend fulfillment system.

For more information on PayPal's Instant Payment Notification, see:

<https://www.paypal.com/us/iph>

PayPal Transaction Reconciliation Report

Although PayPal API calls and Instant Payment Notifications provide an excellent source of "real-time" updates, it can be a good idea to verify that all your transactions have been properly accounted for, rather than relying solely on these methods to track your PayPal transactions. For example, IPNs may be lost due to internet traffic issues, or perhaps the receiving Web server may be down or overburdened, causing you to miss transactions.

The PayPal *Transaction Reconciliation Report* (TRR) provides another method that you can use to reconcile PayPal transactions. The TRR is a daily report provided by PayPal that summarizes the activity for a given account. The report is available each morning via a secure FTP download.

To ensure you have captured all transactions, download and parse the Transaction Reconciliation Report and look for transactions that are **Complete** in the TRR, but have not been accounted for otherwise (for example, they are not available in your IPN database). Because Transaction Reconciliation Reports have abbreviated transaction information, use PayPal API calls to fill in any gaps. For example, you can make a **GetTransactionDetails** PayPal API call to get complete information about a specific transaction.

For more information on the PayPal reporting tools, see:

<https://www.paypal.com/us/cgi-bin/webscr?cmd=p/xcl/rec/reports-products-outside>

Setting Up the PayPal Sandbox

The PayPal Sandbox (in this document, "the Sandbox" for short) is a tool that allows developers to test applications and explore the PayPal API in an environment that is isolated from PayPal's live Web site. It has a Web site that allows you to prepare your Sandbox account for use, e.g., by setting up dummy buyer and seller accounts. It also responds to PayPal API calls in essentially the same way as the live PayPal site.

Creating a PayPal Sandbox account

Your first step toward using the PayPal Sandbox is to create a Sandbox account. To do this:

- 1 Go to the Sandbox signup page at <https://developer.paypal.com>.
- 2 Click the **Sign Up Now** button.
- 3 Fill in the sign-up form on the following page and click **Sign Up**.

Note: In the **Email** field, do not enter the email address that is associated with your live PayPal account. Use an email address that you do not mind sharing with your co-workers. Thus, if you want to share access to your Sandbox account, you can do so without compromising your security.

- 4 The email address you entered in the sign-up form should receive a confirmation email. When you receive this email, follow the enclosed instructions to confirm your sign-up.

- 5 If you do not receive this email within a few minutes, check to see if it has been blocked or diverted as spam. You may have to retrieve it from a mailbox with a name like **Junk** or **Spam**, or from a special mailbox on your mail server.

Logging in to the Sandbox

To log in to the Sandbox:

- 1 Go to <https://developer.paypal.com>.
- 2 Enter your Sandbox account's email address and password.

The Sandbox will display its home page. From this page you can define buyer and seller accounts, examine simulated email generated by PayPal operations in the Sandbox, and do many other things to set up and manage your Sandbox account.

For more detailed information about creating a Sandbox account, see Chapter 2 of the [PayPal Sandbox User Guide](#).

Using test accounts

To test an application in the PayPal Sandbox you must create **test accounts**: simulated PayPal accounts that your application can use to perform PayPal transactions. You typically need at least one seller test account and one buyer test account. Your application's testing requirements may oblige you to create additional test accounts.

You can use the links in the **Test Accounts** section of the Sandbox home page to create, modify, and delete test accounts.

There are two links for creating a test account. One, [Creates a preconfigured buyer or seller account](#), creates an account with many account properties set to commonly used default values. The other, [Manually create accounts](#), gives you substantially complete control of the account's properties.

For more detailed information about creating test accounts, see Chapter 3 of the [PayPal Sandbox User Guide](#).

How to create a preconfigured test account

To create a preconfigured test account:

- 1 Log in to the PayPal Sandbox at <https://developer.paypal.com>, if necessary.
- 2 From the Sandbox home page, click the [Create a preconfigured buyer or seller account](#) link.
- 3 The Sandbox displays a "Create a Sandbox Test Account" form.
- 4 From the form's **Country** list, select the country in which you want the test account to be located.
- 5 From the **Account type** options, select the type of account you want to create: buyer or seller.

- 6 The **Login Email** field contains a **seed** which the Sandbox uses to generate an email account name for the test account. The seed is initially the account name of the email address associated with your Sandbox account. You may enter a different seed if you prefer.

Note about email addresses: The Sandbox does not send real emails; it creates simulated emails which you can only view within the Sandbox itself. Thus you do not need to create real email accounts that correspond to the account names in the Sandbox.

Security note: Do not use the account name of a confidential email account as a seed. For example, do not use your live PayPal account's account name.

- 7 The Password field contains the simulated email account's password. The PayPal Sandbox generates a random password. You may enter a different password if you prefer. Many developers find it convenient to give every test account the same password as their Sandbox account.

Security note: Do not use the password of a real email account (or any other password that protects a "real world" resource).

- 8 In the **Add Credit Card** field, select the type of simulated credit card account that you want to associate with this test account. The Sandbox will generate a random account number for the account.
- 9 Under **Add Bank Account**, click **Yes** or **No** to indicate whether to associate a simulated bank account with this test account. It is recommended that you click **Yes** unless you have a particular reason to do otherwise, as it is difficult to send or receive funds with a test account that has no simulated bank account.

If you click **Yes**, the Sandbox will create a simulated bank account with a random account number and routing number. The routing number will indicate that the bank account is in the country where the test account is located, and the balance will be denominated in that country's currency unit. The simulated bank account will be marked "verified."

- 10 If you are creating a simulated bank account, you can set the **Account Balance** field to give the account a non-zero initial balance. If you will use the account as a buyer, be sure to set an adequate initial balance.
- 11 You can enter text in the **Notes** field, e.g., to document the role of this test account in your testing plan.
- 12 Click the **Create Account** button to create the test account.

After the Sandbox creates the new test account, it displays a list of the test accounts associated with your Sandbox account. Each entry in the list looks something like this:

Log-in Email	Payment Review	Negative Test Mode	Reset
TestSB_1254900954_biz@paypal.com Business <input checked="" type="radio"/>	Disabled	Disabled	<input type="button" value="Reset"/>
<input type="button" value="View Details"/>			

You can click any test account's **View Details** link to display a complete description of that test account.

How to create a test account manually

You may need to create a test account manually if you need to create a test account for a user in a country other than the United States. The Sandbox can create preconfigured test account for users in some other countries, but not all.

To create a test account manually, click the Sandbox home page's [Manually create accounts](#) link. The Sandbox opens a new browser window in which it displays the first page of PayPal's standard account creation procedure. (Note that the URL of this page shows that it is actually a Sandbox page, and will create a Sandbox test account, not a live PayPal account.) You can use the account creation procedure to create a test account with any property settings you need.

A manually created account differs from a live PayPal account in some ways that are natural consequences of the fact that it is not a live account. For example, the Sandbox marks a test account "verified" when you add a bank account to it, but it does not actually verify the account. (You should never give a simulated bank account a real account number or routing code, so verification would always fail if the Sandbox attempted to do it.)

Note on foreign bank accounts. If you need to add a simulated bank account in a nation other than the USA, see the section "Changing or Adding Additional Bank Accounts" in chapter 3 of the [PayPal Sandbox User Guide](#).

For more information about how to create an account manually, see the section "Creating a Test Account Manually" in Chapter 3 of the [PayPal Sandbox User Guide](#).

How to manage test accounts

To manage your test accounts, click any Sandbox page's [Test Accounts](#) tab. (The tabs are on the left side of the page, under the heading [Sandbox](#).) This displays the same page that the Sandbox displays after you define a test account.

To manage a test account, click that account's radio button, then click the [Enter Sandbox Test Site](#) button. The Sandbox opens a new window in which it displays a PayPal log-in page with the test account's email address filled in. Log in to the test account. Now you can manage the test account just as you would manage a live PayPal account.

To delete a test account from the "Test Accounts" summary, click that account's radio button, then click the [Delete](#) button that follows the list of test accounts.

Note that the [Delete](#) button does not delete the test account from your Sandbox account's profile. To do that you must click [Enter Sandbox Test Site](#) and proceed as you would to cancel a PayPal account on the live PayPal site.

To create a new test account, click the [Preconfigured](#) or [Create Manually](#) link above the account summaries. Then follow the appropriate procedure as described above.

How to add funds to a test account

As was explained under "How to create a preconfigured test account," above, you can give a preconfigured test account a non-zero balance when you create it. You can also reset a preconfigured account to its initial balance as explained in "How to reset a test account," below.

You also can add funds to a test account by simulating a transfer from a bank account. This procedure can be used with both preconfigured and manually created test accounts.

To add funds to a test account:

- 1 Log in to the Sandbox.
- 2 Go to the Sandbox's Test Accounts page, select the test account to which you want to transfer funds, and click the [Enter Sandbox Test Site](#) button, as explained in "How to manage test accounts," above.
- 3 Log in to the test account.
- 4 On the account page's [My Account](#) tab, click the [Add Funds](#) link.
- 5 On the following page, click the [Transfer Funds from a Bank Account](#) link.
- 6 On the following page, select the simulated bank from which you want to transfer funds, enter an amount, and click [Continue](#).
- 7 Select the test account page's [History](#) tab.
- 8 Locate the transaction that represents the funds transfer (normally the most recent one). The transaction will show that the funds transfer is "pending." Click the transaction's [Details](#) link.
- 9 Go to the bottom of the following page and click the [Process](#) link. The Sandbox should simulate processing the funds transfer, adding the amount of the transfer to the test account's balance.

How to reset a test account

The Sandbox allows you to reset a preconfigured test account to its original state. This is a convenient way to restore an account to a known state before you rerun a test. Note that you cannot reset a manually created test account.

To reset a test account, go to the Sandbox's Test Accounts page and click the Reset link in that test account's summary. Go to the bottom of the following page and click the Reset button.

When you reset a test account:

- The associated email address and country are preserved.
- The simulated email account's password is set to a new randomly generated number. The test account's balance is set to its initial value, and its Note text is cleared. (You may set the password and note to values of your choice, but you will have to set them again each time you reset the account.)
- All simulated credit card accounts and bank accounts are assigned new randomly generated numbers.

How to read test account email

Certain types of transactions generate email when performed in the live PayPal system. For example, a payment generates a notification email to the payee's account.

In the PayPal Sandbox, such transactions generate simulated email messages that are "sent" and "received" within the Sandbox itself.

These messages are generated from templates by PayPal, so in most cases their content will not concern you. They reflect actions that your application performs; once you confirm that an action is being performed correctly, you may assume that the resulting email is correct.

However, there are situations where it is necessary to read the messages. One is if an email contains details about an order, and you want to be sure that the details not only are correct, but are organized in a way that makes the resulting message easy to read and understand. Another is if you have configured an account manually in a way that causes PayPal to send you an email asking you to confirm your identity. To finish setting up the account you must open the email and go to a URL provided there.

To see emails sent and received by your test accounts, click the [Test Email](#) tab on any Sandbox page. The Sandbox displays the most recent messages sent and received by your test accounts (up to 30 messages). The list of messages will look something like this:

Test Email

Test account email addresses are not real. Email sent to them is never delivered outside the Sandbox.

Below are the most recent messages sent to your Sandbox test accounts.

Inbox

To	From	Subject	Date
jsmith_1279819466_per@ebservice@paypal.com	PayPal electronic funds transfer	Jul. 22, 2010	11:34:17 PDT
jsmith_1279819466_per@ebservice@paypal.com	You have successfully lifted your PayPal withdrawal limit	Jul. 22, 2010	10:24:29 PDT

To see the headers and text of any message, click its subject line.

Note that although PayPal's live system sends messages with formatted HTML text, the simulated messages in the Sandbox are in plain text. To open a URL in a simulated message, you must copy it and paste it into a browser's address field.

Note: Displaying the message in plain text pretty much defeats the stated purpose of making sure that messages with app-supplied text are "easy to read and understand"!

How to use the Instant Payment Notification simulator

Instant Payment Notifications (IPNs) are messages that the PayPal web site sends to an application when the seller account associated with the application receives a payment. IPN messages are typically received and processed by an asynchronous process within the application, called a *listener*.

The IPN simulator is a Sandbox feature which you can use to send simulated IPN messages on demand, without having to perform the transactions that would ordinarily generate them.

To use the IPN simulator:

- 1 Click the [Test Tools](#) tab on any Sandbox page.
- 2 The Sandbox displays the [Test Tools](#) page. Click the [Instant Payment Notification \(IPN\) Simulator](#) link on that page to display the simulator.
- 3 Set the [IPN Handler URL](#) field to the URL to which the simulator should send IPN messages.
- 4 Set the [Transaction type](#) dropdown to the type of IPN you want to send.
- 5 Click the [Send IPN](#) button.

Linking PayPal Test Account to eBay Sandbox

Linking your PayPal test account to an eBay Sandbox account can be helpful in testing Immediate Payment and eBay Stores listings. Use the following procedure to link a PayPal test account to an eBay Sandbox account:

- 1 Log into the eBay Sandbox at: <http://www.sandbox.ebay.com/>.
- 2 From the **My eBay** drop-down menu in the eBay Sandbox, select **Summary**.
- 3 From the **Account** tab in My eBay, select **PayPal Account**.
- 4 Click on **Link My PayPal Account**.
- 5 Enter your PayPal test account credentials.
- 6 Wait for the eBay Sandbox and PayPal test account to link. It generally takes five minutes. You will know that your accounts are linked when the **Account - PayPal Account** page says "*Your eBay account is linked with your Paypal account...*"

Chapter 11

Describing Items in a Listing

As described in “[Listing Items](#)” on page 174, a seller must provide a description of the listing in the **Description** field.

In addition, optional features can help buyers find a listing more easily or make the listing more appealing. For example, a subtitle may help buyers find the listing more easily because it provides more keywords when buyers search in titles and descriptions. Making the listing title bold can help it stand out in the search results gallery.

Some listing features, such as the listing subtitle, have very few special requirements or dependencies. For these, see “[Listing Items](#)” on page 174, along with other Trading API documentation applicable to the features you are using.

Some listing features have requirements or restrictions, or require you to make additional API calls. This section provides details about those.

- “[Introduction to Pictures in Item Listings](#)” on page 206
- “[Working with Pictures in an Item Listing](#)” on page 208
- “[Including Pictures in the Search Results Gallery](#)” on page 213
- “[Specifying an Item’s Condition](#)” on page 218
- “[Offering a Clear Return Policy](#)” on page 223
- “[Enabling Best Offer](#)” on page 225
- “[Enabling Multi-jurisdiction Sales Tax](#)” on page 231
- “[Enabling Get It Fast](#)” on page 233
- “[Identifying Listings that Benefit Nonprofits](#)” on page 234
- “[Identifying Listings that Benefit Nonprofits](#)” on page 234
- “[Listing Items as Private](#)” on page 237
- “[Adult Only Listings](#)” on page 238
- “[Using a SKU Identifier with Items](#)” on page 239
- “[Using Feature Packs to Save on Upgrade Fees](#)” on page 240
- “[Making Listings Available by Default on Another Site](#)” on page 241

In addition to describing the item itself (or items) and providing pictures (at least one picture is required of all listings as of October 1, 2012), sellers will usually want to specify other information, such as shipping details, the available payment methods and other payment instructions, and other useful information related to the selling process itself. Please refer to “[Listing Items](#)” on page 174 and the table of contents for guidance as to where to find such information in this documentation.

Introduction to Pictures in Item Listings

Pictures make an item more appealing, and give buyers a better idea of the item’s appearance. A picture that is stored on the eBay site is stored under eBay Picture Services (EPS). A self-hosted picture is stored on a different site. At least one picture is required for each listing. Up to 12 pictures

can be added to each listing. Pictures are free to include in a listing for all categories except for eBay Motors vehicle listings.

For more information on adding pictures to listings, see:

<http://pages.ebay.com/help/sell/pictures.html>

In addition to the help page above, please see “[Working with Pictures in an Item Listing](#)” on page 208.

The Gallery Plus listing upgrade is a zoom function that allows buyers to roll over a picture and get an enlarged view of the picture. To get the most out of this feature, we recommend that your pictures be 1,000 pixels or larger. Sellers may also want to consider the Value Pack listing upgrade, which includes Gallery Plus, as well as the Subtitle and Listing Designer listing features.

For general information about listing upgrades, see the following help page:

Gallery Upgrades

<http://pages.ebay.com/help/sell/gallery-upgrade.html>

For information on including pictures for an eBay Motors vehicle listing, see the following help page:

Picture Pack

http://pages.ebay.com/help/sell/picture_pack.html

For more information on controlling the display of pictures in a listing, see the description of the **Item.PictureDetails.PhotoDisplay** field in the **AddItem** call reference:

[http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/
AddItem.html#Request.Item.PictureDetails.PhotoDisplay](http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/AddItem.html#Request.Item.PictureDetails.PhotoDisplay)

For the XML API, see the following [UploadSiteHostedPictures](#) topic and Knowledge Base articles:

UploadSiteHostedPictures

<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/UploadSiteHostedPictures.html>

FAQs on eBayPictureServices (EPS) and the UploadSiteHostedPictures API call
https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=1063

Java Sample for UploadSiteHostedPictures

https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=1050

.NET Sample in C# for UploadSiteHostedPictures

https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=1093

PHP Sample for UploadSiteHostedPictures

https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=1092

VB6 Sample for UploadSiteHostedPictures

http://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=1116

For the SOAP API, see the following infomation. The eBay SDK for Java and the eBay SDK for .NET offer a Picture Service library, as described in these *Knowledge Base* articles:

Options for adding pictures to an item (eBay SDK for .NET)

https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=320

Uploading local pictures to EPS using the eBay SDK for Java

https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=723

Working with Pictures in an Item Listing

To associate a picture with an item, specify the picture's URL in the item definition.

In addition to this section, also see “[Introduction to Pictures in Item Listings](#)” on page 206. For information on listing new items, see “[Listing Items](#)” on page 174. For information on retrieving items see “[Retrieving Items](#)” on page 502.

“[Uploading Pictures to EPS](#)” on page 209

“[Associating Pictures with an Item](#)” on page 209

“[Using a Stock Photo in an Item Listing](#)” on page 210

“[Including Pictures in the Description and Using a Camera Icon](#)” on page 211

“[Retrieving Picture Data](#)” on page 212

“[Copy of a Self-Hosted Picture To eBay Picture Services](#)” on page 212

“[Photo Fees](#)” on page 213

Uploading Pictures to EPS

You can use the [UploadSiteHostedPictures](#) call to upload a picture, including a picture located on a web server. This call is available only in the XML API:

UploadSiteHostedPictures Input/Output Topic

<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/UploadSiteHostedPictures.html>

The [UploadSiteHostedPictures](#) call enables you to upload image files by including a binary attachment (or a reference to a picture on a webserver). The call returns [SiteHostedPictureDetails.FullURL](#), which you must store for use in a listing call, when you associate the picture with an item (using the [PictureDetails.PictureURL](#) field).

The [UploadSiteHostedPictures](#) call includes support for watermarks, and returns specific error messages for problems with uploading gallery photos.

If you are using the XML API, please see the following [UploadSiteHostedPictures](#) topic and Knowledge Base article:

UploadSiteHostedPictures Input/Output Topic

<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/UploadSiteHostedPictures.html>

Sample UploadSiteHostedPictures Call

https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=1050

Associating Pictures with an Item

When defining a new item, use fields in [Item.PictureDetails](#) for pictures. The [PictureDetails.PictureURL](#) field specifies a URL for a picture. The [PictureDetails.PhotoDisplay](#) field specifies special effects.

A listing call such as [AddItem](#) will associate pictures with a listing. See the Call Reference for [AddItem](#), which includes descriptions of the [PhotoDisplayCodeType](#) values.

eBay Trading API Call Reference

When you make a call such as [AddItem](#), eBay parses the values you submit in the following [Item.PictureDetails](#) fields: [PictureURL](#) and [GalleryURL](#). eBay determines which pictures are hosted on the eBay site and which are hosted elsewhere (i.e., “self-hosted”). Different rules may apply for self-hosted pictures. However, the size recommendation for a self-hosted picture is the same as for an EPS picture: dimensions more than 1000 pixels on the longer side and quality of 90 or greater for JPEG sources, with a maximum file size of 7 MB.

On a free-Gallery site, if there is a [PictureURL](#), the [PictureURL](#) fills in or replaces the [GalleryURL](#). For the list of free-Gallery sites, see “[Including Pictures in the Search Results Gallery](#)” on page 213.

Also see “[Using Gallery Features on Sites with Free Gallery](#)” on page 215 for information about using Gallery on sites with free Gallery. On free-Gallery sites, if you don’t specify an image in [PictureDetails.GalleryURL](#), the first specified [PictureURL](#) is usually used as the gallery thumbnail (see “[Including Pictures in the Search Results Gallery](#)” on page 213 for exceptions).

You can use `PictureDetails.GalleryType` to specify whether:

- A picture appears in the Gallery
- A picture is Gallery Plus
- A picture is Gallery Featured

See `GalleryTypeCodeType` in the Call Reference for a list of its values.

Note: The Gallery Plus feature (applying to the picture specified in `PictureDetails.GalleryURL` or `PictureDetails.PictureURL`) can be used for images that are self-hosted or that are hosted through eBay Picture Services (EPS). Selecting Gallery Plus does not incur a separate Gallery fee.

For more information about Gallery, Gallery Plus, and Gallery Featured, see “[Gallery, Gallery Plus, and Gallery Featured](#)” on page 213.

The `PictureDetails.GalleryType` value applies to:

- The picture used as the listing’s gallery thumbnail (in `GalleryURL` or `PictureURL`)
- The eBay stock photo, if the stock photo is used as the gallery thumbnail

The `GalleryType` option doesn’t apply to:

- The picture displayed at the top of the listing page
- The pictures embedded in the item description

On non-free-Gallery sites, fees are incurred when `GalleryType` is set to Gallery, Gallery Plus, or Gallery Featured. eBay does not select a picture for the Gallery thumbnail. Instead, eBay uses the picture in `GalleryURL` as the default thumbnail. If you set `GalleryType` to Gallery, but you don’t specify an image in `GalleryURL`, the first specified `PictureURL` is usually used as the gallery thumbnail (see “[Using an eBay Stock Photo as a Gallery Picture](#)” on page 214 for exceptions). If `GalleryType` is set to `None`, the listing contains no Gallery thumbnail.

If Developer Technical Support grants your application an extension of time for picture-hosting, you can extend the hosting time beyond the listing duration; see `ExtendSiteHostedPictures` in the Call Reference.

Using a Stock Photo in an Item Listing

eBay provides stock catalog information, including stock photos, for some commonly sold items. To include a stock photo and other stock product information (also called product details) in an item listing, specify the catalog product ID in your listing request, and then specify additional fields to configure the catalog data you want to use.

Starting in February 2013, a product’s stock photo can only be used for new or refurbished items.

There are three main ways to find a catalog product ID:

- If your item has a UPC, ISBN, EAN, MPN, or you're listing in event ticket categories, you can use [Item.ProductListingDetails](#) to specify an industry-standard value (e.g., an ISBN) or ticket keywords in your listing request, and eBay will use that information to look for a matching product in eBay's catalogs for you.
- If you know of an existing listing on the eBay Web site that was listed with the same product information that you want to use, pass the listing's item ID in [GetItem](#) with a detail level of [ReturnAll](#). This will retrieve the EPID (eBay Product ID) that was used in the listing. Then pass the EPID in [Item.ProductListingDetails.ProductReferenceID](#).
- To search for an EPID, use [FindProducts](#) in the Shopping API. Then pass the EPID in [Item.ProductListingDetails.ProductReferenceID](#) in your listing request.

Once you have identified a catalog product, you can include the product's stock photo in your listing by setting [Item.ProductListingDetails.IncludeStockPhotoURL](#) to true. If a stock photo is available, it is used at the top of the View Item page and in the Item Specifics section of the listing.

If you also include [Item.PictureDetails.PictureURL](#), the stock photo only appears in the Item Specifics section of the listing. Other pictures you specify appear in a separate section of the listing. If you set [IncludeStockPhotoURL](#) to false, the stock photo does not appear in the listing at all. In [GetItem](#) and related calls, the URL of the stock photo will be returned in [StockPhotoURL](#).

Also see “[Using an eBay Stock Photo as a Gallery Picture](#)” on page 214.

If [PictureDetails.GalleryURL](#) and [PictureDetails.PictureURL](#) are not specified (or they do not meet thumbnail requirements), then on sites where Gallery is free and a stock photo is available, eBay uses the stock photo as the gallery thumbnail if you specified [Item.ProductListingDetails.UseStockPhotoURLAsGallery](#) and [Item.ProductListingDetails.IncludeStockPhotoURL](#) as true.

For more information on using a stock photo on sites with free Gallery, see “[Using Gallery Features on Sites with Free Gallery](#)” on page 215.

Including Pictures in the Description and Using a Camera Icon

On non-free-Gallery sites, the Sell Your Item (SYI) page enables a seller to choose the option of providing a picture in the item description. The option is labeled “The description already includes a picture URL for my item” and is in the “Your own Web hosting” tab of the “Add pictures” section of the SYI page. When the listing appears on the search results, a small camera icon appears in the row for the item—even if the item does not have a URL for an image in the [Item.PictureDetails.PictureURL](#) property for the item.

There is no specific [AddItem](#) input field to enable this camera icon. But there is a workaround that for enabling the camera icon for listings that only have an image in the item description. [AddItem](#) only generates the camera icon if an application submits a valid image URL in the [PictureDetails.PictureURL](#) field. But if the only image for the item is embedded in the description, there is no value for [PictureURL](#). To enable the camera icon, provide a specific clear-image URL for [PictureDetails.PictureURL](#). The eBay site will detect this specific clear-image URL and display the

camera icon when the item is displayed in a search page. The URL that must be passed in **PictureDetails.PictureURL**, in **AddItem**, is:

http://pics.ebay.com/aw/pics/dot_clear.gif

The application specifies a picture in the description by including an **** HTML tag in the **Item.Description** field (that is, HTML to display an image embedded in the description), with the source as a URL to the image. For example:

```
<HTML><CENTER><B>eBay API Test Auction: Testing Picture URL icon work-around</B><BR></CENTER> </HTML>
```

The camera icon isn't supported on sites with free Gallery. That is, setting **PictureDetails.PictureURL** to the clear image URL does not enable the green camera icon as the listing's gallery thumbnail on these sites. Note that you can still include a picture in the description section on the listing's page. See "[Introduction to Pictures in Item Listings](#)" on page 206 for more information.

Retrieving Picture Data

Data for pictures is stored in the **Item.PictureDetails** element. The **GetItem** call returns the picture data that was used when the item was listed. For information about the **GetItem** response for the case of a self-hosted picture, see "[Copy of a Self-Hosted Picture To eBay Picture Services](#)" on page 212.

An application should check the **Item.PictureDetails** container to determine picture data. If this element returns a null value, there are no pictures associated with the item. If picture data is returned, the same child elements of **Item.PictureDetails** are used at listing time. (See "[Associating Pictures with an Item](#)" on page 209.)

Copy of a Self-Hosted Picture To eBay Picture Services

On sites with free gallery, if a listing uses a self-hosted picture (except in the case of a multi-variation listing), the picture will be copied to eBay Picture Services (EPS). This copy will be used as the picture at the top of the listing. If the copy fails, the original location is used.

Thus, even in the case of a self-hosted picture, the value returned by **GetItem** in **PictureDetails.PictureURL** is an EPS URL. The original URL is returned in **PictureDetails.ExternalPictureURL**.

When you call **Reviseltem** or **RelistItem**, you can provide a new self-hosted URL in the **PictureURL** field. Do not use the **ExternalPictureURL** input field for this purpose, unless you omit the **PictureURL** field.

The size recommendation for a self-hosted picture is the same as for an EPS picture: dimensions more than 1000 pixels on the longer side and quality of 90 or greater for JPEG sources, with a maximum file size of 7 MB.

Photo Fees

The value returned in the **Fees** object (returned by such calls as **AddItem** and **RelistItem** for newly listed or relisted items) for photos indicates the fee (if any) associated with the use of eBay Picture Services. This service allows you to have one to six photos of the item (more for eBay Motors items) accessible with the item's listing on eBay. The value represents the fees associated with the Photo Service feature that allows a seller to provide photos for the item.

For more information on eBay Picture Services, see the eBay site help page at the URL below.

<http://pages.ebay.com/sell/pictureservices/>

Including Pictures in the Search Results Gallery

Many eBay sites are now free-Gallery sites. Please review the following section: ["Using Gallery Features on Sites with Free Gallery"](#) on page 215. In other cases, when an item is listed, a seller may want the item displayed in the search results Gallery, or use related enhancements. These enhancements are optional and can require additional listing fees.

See the following locations for examples of Gallery, Gallery Plus, and Gallery Featured.

Example of Gallery

http://pages.ebay.com/sell/opt_guide/gallery.html

Example of Gallery Plus

<http://pages.ebay.com/sell/galleryplus-landingpage/>

Example of GalleryFeatured

http://pages.ebay.com/sell/opt_guide/galleryfeat.html

For information on listing new items see ["Listing Items"](#) on page 174 and for information on retrieving items see ["Retrieving Items"](#) on page 502.

Gallery, Gallery Plus, and Gallery Featured

To have a new item appear in the search results Gallery, use the **Item.PictureDetails.GalleryType** property.

Note: Please refer to [Using Gallery Features on Sites with Free Gallery](#) below for information on how to use Gallery features on sites where Gallery is now free. The following sites offer free Gallery: US (site ID 0), Parts & Accessories Category on US Motors (site ID 100), CA (site ID 2), UK (site ID 3), CAFR (site ID 210), FR (site ID 71), NL (site ID 146), ES (site ID 186) and IT (site ID 101). **Gallery, Gallery Plus, and Gallery Featured** applies to sites where Gallery is not free.

GalleryType takes a value of type **GalleryTypeCodeType** (see the type documentation for more on this type). A value of **Gallery** causes the item to be shown in the search results Gallery. A value of **Plus** causes a large image to be displayed on mouseover. A value of **Featured** causes the item to be

highlighted and featured at the top of the Gallery. If no value is specified, then the item is not in the Gallery at all.

Note: Passing Featured, Plus, and/or Gallery together in the same request will return an error.

The **PictureDetails.GalleryURL** property holds a URL for the gallery thumbnail image. If no image is specified in **GalleryURL**, the first image specified in **PictureURL** will be used for the gallery thumbnail in most cases. See “[Using an eBay Stock Photo as a Gallery Picture](#)” on page 214 for exceptions.

Note: The Gallery Plus feature (applying to the picture specified in **PictureDetails.GalleryURL** or **PictureDetails.PictureURL**) can be used for images that are self-hosted or that are hosted through eBay Picture Services (EPS). One EPS gallery image is included with the Gallery Plus fee.

The image used for the Gallery thumbnail must be in one of the graphics formats JPEG, BMP, TIF, or GIF. Non-JPEG formats are converted to JPEG before being displayed in the Gallery. This means the format seen on the eBay Web site will not necessarily be the same as the format specified in **AddItem**. However, the original picture or gallery URL passed in calls to **AddItem** does not change. For example, the URL below is the same specified at listing time as it is when retrieved later—regardless of any image format translations performed by eBay to use the image for the Gallery thumbnail.

`http://www.somedomain.com/images/itempicture.gif`

Using an eBay Stock Photo as a Gallery Picture

eBay provides stock catalog information, including stock photos, for some commonly sold items (e.g., a black iPod Nano). To include a stock photo and other stock information (also called product details) in an item listing, you need to specify the catalog product ID in your listing request, and then specify additional fields to configure the catalog data you want to use.

Note: Please refer to [Using Gallery Features on Sites with Free Gallery](#) below for information on how to use Gallery features on sites where Gallery is now free. The following sites offer free Gallery: US (site ID 0), Parts & Accessories Category on US Motors (site ID 100), CA (site ID 2), UK (site ID 3), CAFR (site ID 210), FR (site ID 71), NL (site ID 146), ES (site ID 186) and IT (site ID 101). [Using an eBay Stock Photo as a Gallery Picture](#) applies to sites where Gallery is not free.

There are three main ways to find a catalog product ID:

- If your item has a UPC, ISBN, EAN, MPN, or you’re listing in event ticket categories, you can use **Item.ProductListingDetails** to specify an industry-standard value (e.g., an ISBN) or ticket keywords in your listing request, and eBay will use that information to look for a matching product in eBay’s catalogs for you.
- If you know of an existing listing on the eBay Web site that was listed with the same product information that you want to use, pass the listing’s item ID in **GetItem** with a detail level of **ReturnAll**. This will retrieve the EPID (eBay Product ID) that was used in the listing. Then pass the EPID in **Item.ProductListingDetails.ProductReferenceID**.
- To search for an EPID, use **FindProducts** in the Shopping API. Then pass the EPID in **Item.ProductListingDetails.ProductReferenceID** in your listing request.

Once you have identified a catalog product, set **GalleryType** to **Gallery**, **Plus**, or **Featured**, as discussed in “[Gallery, Gallery Plus, and Gallery Featured](#)” on page 213. In addition, set both **Item.ProductListingDetails.IncludeStockPhotoURL** and

`Item.ProductListingDetails.UseStockPhotoURLAsGallery` to true. If a stock photo is available, it is used as the gallery picture. The URL of the stock photo will be returned in `StockPhotoURL`.

For sites that don't offer Gallery for free, if the value of `GalleryType` is `Gallery`, `Plus`, or `Featured`, eBay checks the available image URLs in the following order to determine which URL to use as the gallery picture, if more than one of these URLs is available in the listing:

- 1 Use the URL in `GalleryURL`, if it is specified.
- 2 Otherwise, if `ProductListingDetails.UseStockPhotoURLAsGallery` is true, use the eBay stock photo.
- 3 Otherwise, use the value of the first `PictureURL` in the array of `PictureURL` fields, if any.
- 4 Otherwise, if `ProductListingDetails.ProductID` is specified, use the eBay stock photo for that product ID (regardless of the value of `IncludeStockPhotoURL`). eBay resets `UseStockPhotoURLAsGallery` to true in this case.

If no `GalleryURL`, `PictureURL`, or eBay stock photo is found, an error is returned if `GalleryType` is specified with `Gallery`, `Plus`, or `Featured`.

Also see “[Using a Stock Photo in an Item Listing](#)” on page 210.

Using Gallery Features on Sites with Free Gallery

The Gallery feature (i.e. when `Item.PictureDetails.GalleryType` is `Gallery`) is free on the following sites:

- US (site ID 0)
- Parts & Accessories Category on US Motors (site ID 100)
- CA (site ID 2)
- UK (site ID 3)
- CAFR (site ID 210)
- FR (site ID 71)
- NL (site ID 146)
- ES (site ID 186)
- IT (site ID 101)

On a free-Gallery site, if there is a `PictureURL`, the `PictureURL` fills in or replaces the `GalleryURL`. For listings on these sites, eBay can automatically create a gallery thumbnail with a picture from one of the image URLs specified in the request (i.e. `Item.PictureDetails.GalleryURL` or `Item.PictureDetails.PictureURL`).

eBay selects an image depending on which of these image URLs was specified (four cases) and whether a stock photo is available for the item. eBay can also create a gallery thumbnail for the listing even if `Item.PictureDetails.GalleryType` is not specified or is set to `None`. Fees will only apply whenever

PictureDetails.GalleryType is **Plus** or **Featured** (no fee whenever **GalleryType** is **None** or is not included in the request).

On sites that offer free Gallery, enhancements that were made in **Item.PictureDetails.GalleryType** can be downgraded, upgraded, or removed when using **RelistItem** or **Reviseltem**. If a seller tries to downgrade a **GalleryType** enhancement using a **GalleryType** value of **None**, then the request is ignored.

The intent of eBay's selection rules is have the gallery thumbnail match the picture that appears at the top of the listing's page (which is specified by the first **PictureDetails.PictureURL**) whenever possible, as long as this picture is valid. As an example, if both **PictureURL** and **GalleryURL** have been included in the request, eBay uses the picture referenced by **PictureURL** as the gallery thumbnail (copying this URL into **GalleryURL**). It then applies any options specified by **GalleryType** (i.e., **Plus** or **Featured**). As a result, both the gallery thumbnail in the search results and the picture at the top of the listing's page now match, providing a more consistent experience for buyers when browsing from search results to listing pages.

eBay's picture selection rules on sites with free gallery always attempt to find a picture for the gallery thumbnail and synchronize it with the picture displayed at the top of the listing's page.

An image is eligible to be used as the gallery thumbnail if it is not the null-image URL located at http://pics.ebay.com/aw/pics/dot_clear.gif and is greater than 20 pixels in both length and width.

For specifying pictures in your listing on sites with free gallery:

- To display a picture as the gallery thumbnail and at the top of the listing, set the first **PictureURL** field to a valid (i.e., gallery-eligible) image and omit **GalleryURL**.
- To display an eBay stock photo as the gallery thumbnail and at the top of the listing, omit all **PictureURL** elements in the request. You must set **UseStockPhotoURLAsGallery** and **IncludeStockPhotoURL** to true. Specifying **GalleryURL** as a backup image is a good practice in this case, as it prevents your listing from displaying a blank thumbnail in the search results in the event that the catalog product you are listing with doesn't have a stock photo.

Note that for free-gallery sites, there is no way to specify different images as the gallery thumbnail and as the picture at the top of the listing. If eBay can select a picture for the gallery thumbnail, eBay displays the same picture in both places (and **GetItem** returns the same image URL in both **GalleryURL** and the first **PictureURL**).

The following describes four cases for **PictureDetails.GalleryURL** and **PictureDetails.PictureURL**. Note that the first **PictureURL** is valid for use as the gallery thumbnail only if it is not the null-image URL at http://pics.ebay.com/aw/pics/dot_clear.gif and is greater than 20 pixels in both width and height. For each case below, eBay checks the image URLs in the specified order to determine the image for the gallery thumbnail.

Both `GalleryURL` and `PictureURL` are specified in the request:

- 1 eBay uses the first `PictureURL`, if it is valid for use as a gallery picture. The value in `GalleryURL` is replaced with this URL. Thus, `GetItem` returns a response where `GalleryURL` equals the first `PictureURL`.
- 2 Otherwise, eBay uses the URL in `GalleryURL`. The value in the first `PictureURL` is replaced with this URL.

Note that `Item.ProductListingDetails.UseStockPhotoURLAsGallery` is ignored in the request for this particular scenario. You can still use `Item.ProductListingDetails.IncludeStockPhotoURL` in the request with expected results.

Note: On these sites, the green camera icon no longer displays as the gallery thumbnail for listings that have set the first `PictureURL` to the null-image URL. If a picture for the Gallery thumbnail is not found, a blank thumbnail (without a camera icon) is displayed in the search results. However, you can still embed images within the `Item.Description` field in the listing request. See [Including Pictures in the Description](#) and [Using a Camera Icon](#) for more information about embedding images into the listing page's Description section.

Only `GalleryURL` is specified in the request:

- 1 If you specified `Item.ProductListingDetails.UseStockPhotoURLAsGallery` and `Item.ProductListingDetails.IncludeStockPhotoURL` as true, then if a stock photo is available, both `PictureURL` and `GalleryURL` are set to the stock photo's URL.
- 2 Otherwise, eBay uses the URL in `GalleryURL`, and the first `PictureURL` is set to this URL.
- 3 If you did not specify `Item.ProductListingDetails.UseStockPhotoURLAsGallery` and `Item.ProductListingDetails.IncludeStockPhotoURL` as true, no Gallery thumbnail is generated.

For information about using catalog data, see “[Using a Stock Photo in an Item Listing](#)” on page 210.

Only `PictureURL` is specified in the request:

- 1 eBay uses the first `PictureURL`, if it is valid. `GalleryURL` is set to this URL. You can still set `Item.ProductListingDetails.IncludeStockPhotoURL` as true in the request, and a stock photo will be included if available. Only `Item.ProductListingDetails.UseStockPhotoURLAsGallery` is ignored if included.
- 2 Otherwise, a blank thumbnail displays in the search results for this listing. Note that if either `Item.ProductListingDetails.UseStockPhotoURLAsGallery` or `Item.ProductListingDetails.IncludeStockPhotoURL` are included as true, a stock photo will be used as the gallery thumbnail as well as the picture displayed at the top of the item's listing page

Neither `GalleryURL` nor `PictureURL` are specified in the request:

- 1 If you specified `Item.ProductListingDetails.UseStockPhotoURLAsGallery` and `Item.ProductListingDetails.IncludeStockPhotoURL` as true, then if a stock photo is available, both `PictureURL` and `GalleryURL` are set to the stock photo's URL.
- 2 If you did not specify `Item.ProductListingDetails.UseStockPhotoURLAsGallery` and `Item.ProductListingDetails.IncludeStockPhotoURL` as true, no Gallery thumbnail is generated.

Specifying an Item's Condition

Item condition is one of the top factors buyers consider when deciding whether to purchase an item. When you create your listing, help your buyers understand exactly what they'll be receiving by specifying its condition. Listings that specify the item's condition provide useful information and as a result, may be more likely to sell.

Improve Performance with Better Descriptions (eBay online help)
<http://pages.ebay.com/help/sell/descriptions.html>

Many eBay sites support the ability to specify the condition of an item in a standardized, structured way (as a name/value pair). In most categories, eBay defines a list of item conditions that you can choose from.

Your item's condition appears at the top of your listing description. You can provide clarifications about the condition within the listing description.

When buyers search for items, they often use eBay's Refine Search options to narrow their search based on the item's condition. In these cases, your listing will appear in search results only if you specified the condition by using one of eBay's structured methods (not in the text of your listing description).

This section covers these topics:

["Using ConditionID to Specify an Item's Condition" on page 218](#)
["Describing Details About the Condition of Items That Are Not New" on page 221](#)
["Adding or Changing ConditionID When Revising or Relisting" on page 222](#)
["Retrieving a Listing's Item Condition" on page 223](#)

Using ConditionID to Specify an Item's Condition

For most categories, eBay defines a list of well-understood conditions, like "New", "Used", and other values. Each condition has a numeric ID (like 1000 for "New") so that even if the display name varies across categories or sites, you can use the same IDs in `AddItem` and related calls.

To find the right condition ID to use in your listing:

- 1 Use `GetCategoryFeatures` to determine whether `ConditionEnabled` is set to `Enabled` or `Required` for the category (or its ancestors).
- 2 If this condition model is enabled (or required), look up the list of valid condition IDs in `ConditionValues`.

Each ID also has a human-readable display name, such as "New" or "Manufacturer refurbished". These are returned in `GetItem`, but you cannot pass them in `AddItem`.

- 3 After choosing the most accurate condition ID from `GetCategoryFeatures`, use `Item.ConditionID` to specify the condition in `AddItem` (or related calls).
- 4 If the item is not new, consider also adding `ConditionDescription` to describe any wear and tear. See ["Describing Details About the Condition of Items That Are Not New" on page 221](#).

Please note the following behavior if you pass a **ConditionID** value that is not valid for the category:

- If **ConditionEnabled** is disabled (or not applicable) for the category, the item is listed with no condition.
- If **ConditionEnabled** is set to **Enabled** or **Required** for the category, the request fails.

We recommend that you read these sub-topics to understand more about how to work with item conditions, and how to decrease disputes with buyers:

- ["Item Condition for US eBay Motors" on page 219](#)
- ["Helping Sellers Choose the Right Condition" on page 219](#)
- ["Listing in Two Categories" on page 219](#)
- ["Condition Differences Across Categories" on page 220](#)

Item Condition for US eBay Motors

The US eBay Motors site supports **ConditionID** for vehicles as well as parts & accessories. Call **GetCategoryFeatures** to determine which categories require **ConditionID** for newly listed or re-listed items.

Helping Sellers Choose the Right Condition

Although many categories use the same condition IDs, buyers may interpret the conditions differently in each category. It is very important that sellers understand what each condition means for each category, because it is part of the contract between the buyer and the seller. So, **GetCategoryFeatures** also returns links to eBay's online help (**ConditionHelpURL**).

We recommend that you display the help links prominently in your application interface, so that sellers understand how buyers will interpret the item condition. (Or, if you are mapping older condition values to the new field on behalf of sellers, you should be aware of eBay's definitions for each condition.)

Note: If a category (such as one in Antiques or Collectibles) doesn't support **ConditionID**, but you still want to describe the item's condition in your listing, here are some options:

If the category supports custom item specifics but doesn't specifically recommend Condition as a name, make up your own custom item specific for the condition. Alternatively, explain the condition in your item description.

In these cases, eBay's search engine will treat your text like any other keyword. That is, eBay will not recognize or process the item's condition when users filter on condition, but your text will be searchable and visible to buyers when they view the listing.

Listing in Two Categories

If you are listing in two categories, the primary category determines which condition model (**ConditionID** or item specifics) and which condition values can be used.

On the eBay site, you may see items that appear to have an item condition specified in **ConditionID** even though **GetCategoryFeatures** shows that condition is disabled for the category. This can occur when the seller lists in two categories, because the primary category's rules take precedence.

So, for example, on the eBay US site, the Technology Books & Resources (3516) category supports **ConditionID**, but the Vintage Computing (11189) category doesn't. So, as long as a listing uses 3516 as the primary category ID and 11189 as the secondary category ID, the listing can use any of the **ConditionID** values that category 3516 supports.

Condition Differences Across Categories

In general, each meta-category (level-1 category) defines its own default set of conditions (rather than inheriting from a site default). Most meta-categories on a given site will actually define the same default set of conditions, but some meta-categories define different default sets to meet buyer and seller expectations. Furthermore, within each meta-category, a significant number of child categories may use different sets of conditions than their parent categories. Use [GetCategoryFeatures](#) to determine which conditions each category supports.

For example, while the Sporting Goods category on the eBay US site might define a default set of conditions that most of its child categories will inherit, a child category like Running > Shoes would use the same set of conditions as the Clothing, Shoes & Accessories > Shoes category instead.

Similarly, the Books meta-category and meta-categories where items are typically CDs or DVDs (i.e., the DVD & Movies, Music, and Video Games meta-categories) are considered “media categories”, and they typically all define the same default set of conditions. But a child category like Video Games > Systems has items that are electronic devices, so it defines a different set of conditions (the same set as the Computers & Networking meta-category).

Table 11-1 on page 220 lists the conditions that are supported across the eBay US site, plus conditions that some meta-categories and child categories actually use.

Please note these details about the data in the table:

- We consistently use the same IDs for similar condition qualities (across all sites). For example, 1000 is always used for brand new items in their original retail packaging (or equivalent). However, the display names and the number of conditions can vary in different sets. Also, not all sites support equivalent values. For example, an EU site might support condition ID 3000 (Used) for Books, whereas the US site supports conditions like 2750 (Like New) and 4000 (Very Good) for Books instead.
- If no display name is shown, it means the ID isn't supported in the category.

Table 11-1 Examples of Conditions in eBay US Categories

ID	Typical Name	Video Games (1249)	Video Games > Systems (139971)	Shoes (163450)
1000	New	Brand New	New	New with box
1500	New other (see details)		New other (see details)	New without box
1750	New with defects			New with defects
2000	Manufacturer refurbished		Manufacturer refurbished	
2500	Seller refurbished		Seller refurbished	
2750	Like New	Like New		
3000	Used		Used	Pre-owned
4000	Very Good	Very Good		
5000	Good	Good		
6000	Acceptable	Acceptable		
7000	For parts or not working		For parts or not working	

eBay doesn't define conditions in categories where an item condition make no sense or would never be anything other than new. For example, an item condition would not apply to Event Ticket listings.

Note: eBay defines 10-15 overall sets of conditions. Each site may have a few hundred categories that use one of these sets instead of a meta-category's default set. The condition sets are determined based on testing with sellers and category experts, as well as research into what buyers expect.

Describing Details About the Condition of Items That Are Not New

Starting in October 2012, eBay will support a new field called Condition Description (in the Sell Your Item form and other tools) to help sellers more clearly describe the condition of items that are not brand new.

The condition description field will be available for all conditions except new, brand new, new with tags, and new in box. This field also applies to categories where the condition type is not applicable (e.g., Antiques).

The condition description field will help buyers more quickly find and understand an item's condition and to reduce disputes, as many buyers miss comments that are buried deep in the seller's item description.

AddItem and related calls will support a new **ConditionDescription** field for this functionality. We recommend that you start using it when available, to help buyers better understand the condition of items that are not new.

GetItem and related calls in the Trading API, and **GetSingleItem** and **GetMultipleItems** in the Shopping API will return **ConditionDescription**.

The Finding API and other search calls will not return condition descriptions. eBay's search engine will not search text within **ConditionDescription**.

We are aiming to release this functionality in the Sandbox in September, and in production in October. Keep an eye on the release notes and announcements from eBay and x.com for updates.

Policies and Best Practices for Sellers When Using Condition Description

Please keep the following policies and best practices in mind when you create listings:

- Choose the item condition type that most accurately describes the wear and tear on your item.
- Use the new condition description field to further clarify the condition of the item. For example, "The right leg of the chair has a small scratch, and on the seat back there is a light blue stain about the shape and size of a coin."
- Do not use the condition description for branding, promotions, shipping, returns, payment or other information unrelated to the condition. If you do, the listing request may fail or the listing may be administratively removed later.
- Make sure the condition type, condition description, pictures, and item descriptions for a listing do not contradict one another.

To avoid contradictory or confusing descriptions:

- Move condition details out of your item description and into this new Condition Description field, once it becomes available.
- Do not duplicate content in both the item description and the condition description, as this may confuse buyers, or your listing request may fail.
- Do not split the condition details between the item description and the condition description. For example, don't partially describe the condition in the Condition Description field and add "See description" for the rest of the details, as this may confuse buyers and lead to disputes.

Rules for Using ConditionDescription in Listing Requests

The following details will be available in the Call Reference for [AddItem](#) and related calls in September. They are being provided here for early reference:

- **ConditionDescription** was an optional field in 2012. It may become required for some categories in 2013, with Seller Release 13.1.
- **ConditionDescription** will be available for all eBay categories on all sites, including categories that do not currently support **ConditionID** (such as Antiques).
- **ConditionDescription** can be used when you pass in **ConditionID** values of 1500 and higher, or when no **ConditionID** is specified (e.g., for categories like Antiques, which do not support **ConditionID**).
- For categories that support **ConditionID**, eBay ignores (drops) the **ConditionDescription** field when **ConditionID** is 1000-1499 (e.g., new, brand new, new with tags, new in box), as the description is not relevant to new items.

Note that if the **ConditionDescription** is dropped, a warning message is returned. If your application provides a user interface (or a summary of warning messages to sellers who list in bulk), it may be helpful to display this message, in case the seller has a good reason to clarify the condition of the new item in the item description instead.

- In [Reviseltem](#) and [ReviseFixedPriceltem](#), you cannot modify **ConditionDescription** when an auction listing has bids or ends within 12 hours, or when a fixed-price listing has outstanding best offers.
- In the initial release, **ConditionDescription** will always show the latest string the seller entered, even for past purchases. (In a future release, the **ConditionDescription** will retain its original text for purchased items.)
- The maximum length of a condition description is 1000 characters.

Adding or Changing ConditionID When Revising or Relisting

To add **ConditionID** to an existing listing, specify it in [Reviseltem](#), [RelistItem](#), and related calls in the usual way. See "[Using ConditionID to Specify an Item's Condition](#)" on page 218.

In most cases, you can add or modify **ConditionID** when you revise or relist. If **GetCategorySpecifics** specifies **ConditionEnabled** as **Required** for a category, you can change the ID to more accurately describe the item, but you cannot remove **ConditionID** from the listing when you revise or relist it.

Modifying ConditionID for Listings in Restricted State

If the listing is in a restricted state, the condition is subject to eBay's usual revision restrictions. If you attempt to add or change **ConditionID** when a listing is in a restricted state, your request may fail.

For example, if an auction has bids or ends within 12 hours, you cannot remove or change its condition. In this case, if there are fields that are still editable on the auction, you will be able to edit them (even though **ConditionID** is not present).

See "[Item Changes Allowed During Revisions](#)" on page 382 for typical rules about fields that can't be modified when an item is in a restricted state.

Retrieving a Listing's Item Condition

GetItem returns the item condition in whichever field the seller used to specify it. Check **ConditionID** or **ItemSpecifics** to find the condition.

For items listed with **ConditionID**, **GetItem** also returns the human-readable condition label in **ConditionDisplayName**. The label is always in the language of the site on which the item was originally listed. That is, if a seller listed a brand new item on the eBay Germany site, and then you call **GetItem** but you send the request to the eBay US site, **ConditionDisplayName** will still return "Neu" in German (it will not convert the display name to "New").

You can use calls like **findItemsAdvanced** in the Finding API to retrieve the new **ConditionID** field in searches.

Offering a Clear Return Policy

eBay's research shows that "difficulty in returning items" is the top shopping barrier cited by buyers. Often, the source of the difficulty is buyer confusion over a seller's return policy. This means sellers who clearly spell out a return policy in their listings have an edge on the competition. That is, sellers who provide a clear return policy are likely to sell a higher percentage of their listed items than sellers who do not have a clear return policy.

eBay's research also shows that only a very small percentage of sold items are actually returned.

Due to differences in legal requirements for each country, different sites support different return policy options. To make return policies easier to specify and to understand, most eBay sites offer a standardized way to specify and display return policy information.

Sellers are required to specify a return policy for auction and fixed-price listings in most categories. Sellers can choose not to accept returns, but they must still specify this clearly by using the standard return policy options.

Note: Regardless of your stated return policy, all sellers are expected to adhere to eBay's Rules for Sellers, which include accepting a returned item when the item is not as described.

Rules for Sellers (eBay US)

http://pages.ebay.com/help/policies/ia/rules_for_sellers.html

For general information, rules, and legal guidance on offering a return policy, see the eBay Web site's online help. Here are a few examples:

Specifying Your Return Policy (eBay US)

<http://pages.ebay.com/help/sell/your-return-policy.html>

Ihre Rücknahmebedingungen (eBay Germany)

<http://pages.ebay.de/help/sell/your-return-policy.html>

Legal Guidance for Business Sellers (eBay UK)

<http://pages.ebay.co.uk/help/policies/business.html>

Returns & the Law (eBay UK)

<http://pages.ebay.co.uk/businesscentre/law-policies/returns.html>

Legal Guidance for Business Sellers (eBay Ireland)

<http://pages.ebay.ie/help/policies/business.html>

Returns & the Law (eBay Ireland)

<http://pages.ebay.co.uk/businesscentre/law-policies/returns.html>

Specifying Your Return Policy (eBay Australia)

<http://pages.ebay.com.au/help/sell/your-return-policy.html>

Specifying Your Return Policy (eBay India)

<http://pages.ebay.in/help/sell/your-return-policy.html>

Guidelines for Creating Legally Compliant Listings (eBay India)

<http://pages.ebay.in/help/tp/compliant-listings.html>

Warranties (eBay India)

<http://pages.ebay.in/help/policies/warranties.html>

Listing an Item with a Return Policy

To include your return policy in a listing:

- 1 For the site you're listing on, use [GetCategoryFeatures](#) to determine which categories require a return policy (see [ReturnPolicyEnabled](#)).
If either the primary or secondary category in your listing requires a return policy, then the listing must include a return policy.
- 2 For the site you're listing on, use [GetBayDetails](#) to determine which return policy fields are supported, and which values are supported (see [ReturnPolicyDetails](#)).
- 3 If your application has a user interface, present the available return policy options.
Buyers and sellers typically expect return policy information to be grouped with payment and shipping instructions.
- 4 When you list the item by using [AddItem](#) and related calls, specify the return policy options in the [Item.ReturnPolicy](#) fields.

Please note that [ReturnPolicyType](#) (which is the type used for [Item.ReturnPolicy](#)) defines a pair of fields for several return policy options. For example, for the refund option, there is a [RefundOption](#) field (an enumeration of type `xs:token`) and a [Refund](#) field (a localized display string). You can only use the "option" field (e.g., [RefundOption](#)) in [AddItem](#) and related calls. You can't use the other field (e.g. [Refund](#)) in these calls. The localized string is only provided as a convenience for displaying refund options in user interface elements.

Using the [>ReturnsWithin](#) (text field) or [>ReturnsWithinOption](#) (token value based on [>ReturnsWithinOptionsCodeType](#)), the seller must state the number of days a buyer has to return or exchange the item. As of May 2012, 3- and 7-day return policies are no longer be allowed by the US site and many other sites. To verify what [>ReturnsWithinOption](#) values are allowed for a specific site, the seller should call [GetBayDetails](#), using the [ReturnPolicyDetails](#) as a input filter.

Using the [Refund](#) (text field) or [RefundOption](#) (token value based on [RefundOptionsCodeType](#)), the seller must state their refund policy (e.g., Money Back). As of May 2012, the [MerchandiseCredit](#) and [Exchange](#) values in [RefundOptionsCodeType](#) are no longer be allowed by the US site. The US seller's only two refund options will be [MoneyBack](#) or [MoneyBackOrExchange](#).

Also see "[Item Changes Allowed During Revisions](#)" on page 382.

Retrieving an Item's Return Policy

[GetItem](#), [GetSellerList](#), and [GetBidderList](#) all return the [Item.ReturnPolicy](#) fields. For your convenience, they return both the enumeration values that can be used in [AddItem](#) as well as localized display strings for each field.

Enabling Best Offer

A seller of a fixed price format item (in a category for which Best Offer is also enabled) can opt that item into the Best Offer feature. This feature enables a buyer to make a lower-priced, binding offer on that item. That is, the Best Offer feature enables a buyer to make a best offer on an item.

The seller can accept the offer, submit a counteroffer, decline the offer, or let the offer naturally expire after 48 hours. If the seller accepts an offer, the listing ends and the Best Offer price and terms are displayed on the View Item page.

If a seller enables an item for best offer, and a buyer wants to make a best offer (and a follow-up counteroffer), this is supported through the API (see “[PlaceOffer](#)” on page 904) and through the eBay site.

The maximum number of best offers allowed for an item is configurable by eBay, per site and per category, and can change without notice as market needs change. (The maximum number could be as low as 1 and as high as 10 or more, depending on the category.) Therefore, if a buyer makes an offer on a particular item, and the seller declines the offer, the configuration for the category dictates whether the buyer can make a follow-up offer on the item.

A buyer is alerted when making the final offer allowed for an item, but a buyer is not told how many offers are permitted for items in the category. This is intentional, to discourage buyers from making less-than-serious offers for an item. A [BestOffer](#) SOAP notification is created when a buyer makes a best offer on an item for which Best Offer is enabled. See “[BestOffer](#)” on page 642.

A seller cannot know how many offers a buyer can make per item and should consider each offer submitted as the buyer’s Best Offer. If an offer does not meet the seller’s minimum sale price, the seller can decline the best offer or submit a counteroffer. The seller uses [RespondToBestOffer](#) to accept or decline a best offer or to make a counteroffer.

Offers declined by the seller, retracted by the buyer, or allowed to expire count against the buyer’s maximum number of offers allowed per item per site.

Counteroffer Rules and Restrictions

If a seller makes a counteroffer, the buyer can accept the counteroffer, decline the counteroffer, or allow the counteroffer to expire. In some instances, buyers can make counteroffers to the seller’s counteroffer.

Note these additional rules and restrictions for counteroffers:

- For buyers, counteroffers count against the maximum number offers allowed per item per buyer based on the category configuration. If the site and category allow multiple offers per item, the

buyer can make a counteroffer to the seller's counteroffer. This can continue until the buyer makes the maximum allowed number of offers.

- When a buyer declines a counteroffer, it does not count against the buyer's maximum number of offers.
- Sellers can make counteroffers only to buyers who have active offers or counteroffers.
- If the seller has an active counteroffer out to the buyer, the sellers cannot send another counteroffer to the buyer until the buyer counters back to the seller.
- Sellers cannot make counteroffers above the Buy It Now price for items unless buyers first submit best offers above the Buy It Now price.
- If the buyer declines a counteroffer, the seller cannot initiate a new counteroffer to the buyer unless and until the buyer initiates a new best offer.
- When a counteroffer is submitted, all previous offers are declined. That is, after extending a counteroffer to a buyer, the seller will not be able to accept a previous offer from that buyer for that item. Similarly, after sending a counteroffer back to the seller, the buyer cannot accept a previous counteroffer from the seller.
- Counteroffers, like best offers, expire in 48 hours for both buyers and sellers. When the seller sends a buyer a counteroffer, the buyer's original offer time is cleared. The buyer now has 48 hours to accept the new offer from the seller.
- Sellers are able to counteroffer with multiple buyers simultaneously.

Single Quantity Counteroffers

The first buyer to accept the counteroffer wins the item. All non-winning best offers are automatically declined. If the item was sold using Buy It Now, all offers are declined because the item has ended.

Multi-quantity Counteroffers

For multi-quantity listings, sellers are able to make counteroffers to multiple buyers, and each offer that is accepted decrements inventory. Each individual counteroffer submitted to a buyer cannot exceed the inventory the seller has available.

Sellers cannot submit a counteroffer for a higher quantity than the buyer initially submitted, but buyers can counter for a quantity higher than the seller's counteroffer.

Retracting Counteroffers

Buyers and sellers can retract counteroffers through the site. Since creating a counteroffer essentially declines the previous offer, when a counteroffer is retracted, the offer thread is no longer active. That is, retracting a counteroffer does not reactivate the previous offer.

Automatically Declining or Accepting an Offer

When an item is listed with the Best Offer feature, the seller can specify a minimum best offer price, a minimum best offer message, and an auto-accept price.

Use [GetCategoryFeatures](#) to determine if these features for best offer are supported for the category you want to use.

If you set a minimum best-offer price and an offer does not meet it, the offer is automatically declined and the minimum best offer message is sent to the offeror. The minimum best offer price and minimum best offer message are specified with the [Item.ListingDetails.MinimumBestOfferPrice](#) and [Item.ListingDetails.MinimumBestOfferMessage](#) fields when items are listed, relisted, or revised with the following calls: [AddItem](#), [ReviselItem](#), [RelistItem](#), and [VerifyAddItem](#). A minimum best offer price or message can only be specified for items listed in categories that support auto-decline for best offers. If a minimum best offer price and message have been specified for a listing in a category that supports auto-decline for best offers, [GetItem](#) and [GetSellerList](#) will return the [Item.ListingDetails.MinimumBestOfferPrice](#) and [Item.ListingDetails.MinimumBestOfferMessage](#) fields.

The auto-accept price is the price at which best offers are automatically accepted. Use of an auto-accept price is similar to use of a minimum best offer price. Specify an auto-accept price as [Item.ListingDetails.BestOfferAutoAcceptPrice](#) when an item is listed, relisted, or revised with the following calls: [AddItem](#), [ReviselItem](#), [RelistItem](#), and [VerifyAddItem](#).

Anonymous Best Offer

Anonymous Best Offer is an enhancement to the Best Offer feature. For items listed as Best Offer items on sites for which Anonymous Best Offer is enabled, the response logic for the [RespondToBestOffer](#) and [GetBestOffers](#) calls is altered to conceal the identity of buyers making offers on an item.

GetBestOffers Changes

On sites that have Anonymous Best Offer, the buyer's user ID ([BestOffer.Buyer.UserID](#)) is obscured in the [GetBestOffers](#) response by replacing the last three characters with asterisks (*). Additionally, the following three fields are not returned:

- [BestOffer.Buyer.User.Email](#)
- [BestOffer.BuyerMessage](#)
- [BestOffer.SellerMessage](#)

The following sample shows a typical [GetBestOffer](#) response for a site that does not support Anonymous Best Offer:

Example 11-1 Typical GetBestOffer response

```
<GetBestOffersResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  ...
  <BestOfferArray>
    <BestOffer>
      <BestOfferID>7172</BestOfferID>
      <ExpirationTime>2006-01-22T07:36:35.000Z</ExpirationTime>
      <Buyer>
        <Email>username@hostname.co.uk</Email>
        <RegistrationDate>1999-01-19T08:00:00.000Z</RegistrationDate>
        <UserID>username</UserID>
      </Buyer>
    </BestOffer>
  </BestOfferArray>
</GetBestOffersResponse>
```

Example 11-1 Typical GetBestOffer response (Continued)

```
<Price currencyID="USD">40.00</Price>
<Status>Pending</Status>
<Quantity>1</Quantity>
<BestOfferCodeType>BuyerBestOffer</BestOfferCodeType>
<BuyerMessage>
    Message from the buyer to the seller.
</BuyerMessage>
</BestOffer>
</BestOfferArray>
<Item>
    <BuyItNowPrice currencyID="USD">45.0</BuyItNowPrice>
    <ListingDetails>
        <EndTime>2005-12-22T06:30:47.000Z</EndTime>
    </ListingDetails>
</Item>
</GetBestOffersResponse>
```

On a site with Anonymous Best Offer, the response would not include the buyer email address or the buyer message, and the user ID for the buyer would be obscured, as show in the following sample:

Example 11-2 GetBestOffer response with Anonymous Best Offer

```
<GetBestOffersResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    ...
    <BestOfferArray>
        <BestOffer>
            <BestOfferID>7172</BestOfferID>
            <ExpirationTime>2006-01-22T07:36:35.000Z</ExpirationTime>
            <Buyer>
                <RegistrationDate>1999-01-19T08:00:00.000Z</RegistrationDate>
                <UserID>usern***</UserID>
            </Buyer>
            <Price currencyID="USD">40.00</Price>
            <Status>Pending</Status>
            <Quantity>1</Quantity>
            <BestOfferCodeType>BuyerBestOffer</BestOfferCodeType>
        </BestOffer>
    </BestOfferArray>
    <Item>
        <BuyItNowPrice currencyID="USD">45.0</BuyItNowPrice>
        <ListingDetails>
            <EndTime>2005-12-22T06:30:47.000Z</EndTime>
        </ListingDetails>
    </Item>
</GetBestOffersResponse>
```

At the time of the 445i international release, Anonymous Best Offer was enabled on the following sites:

- Austria (<http://www.ebay.at/>)
- Germany (<http://www.ebay.de/>)
- Spain (<http://www.ebay.es/>)
- Switzerland (<http://www.ebay.ch/>)
- United Kingdom (<http://www.ebay.co.uk/>)

Sellers' Use of Best Offer

The following steps outline a seller flow for the Best Offer feature:

- 1 Check if a category supports Best Offer.

To determine if a category supports Best Offer listings, check `CategoryArray.Category.BestOfferEnabled` in the response of [GetCategories](#).

- 2 Enable Best Offer for your listing.

To opt an item into Best Offer, set `Item.BestOfferDetails.BestOfferEnabled` to true when listing, revising or relisting the item.

- 3 Verify the Best Offer settings for the item.

To determine whether Best Offer is enabled for an item and how many best offers exist for it, check `BestOfferEnabled` and `BestOfferCount` in `Item.BestOfferDetails` in the response from [GetItem](#).

- 4 Get information about any best offers made for the item.

Use [GetBestOffers](#) to get information about best offers made for an item.

- 5 Respond to a best offer by accepting, declining, or making a counteroffer.

Use [RespondToBestOffer](#) to accept or decline a best offer or to make a counteroffer. Depending upon the site and category limitations on the number of offers a buyer can make, the buyer can accept the counteroffer, decline the counteroffer, or submit a counteroffer to the seller's offer. Eventually, the buyer must accept, decline, or allow the seller's last offer to expire.

- 6 When a best offer has been accepted, check on the status of the order line item.

To determine if a order line item was completed with the seller selecting a best offer, check `Transaction.BestOfferSale` in [GetItemTransactions](#).

Note that [GetSellerList](#) returns `ItemArray.Item.BestOfferDetails.BestOfferEnabled`. [GetBidderList](#) has no Best Offer component.

Additional Help for Best Offer

You can also find discussion of Best Offer in the Help section of the eBay Web site.

<http://pages.ebay.com/help/sell/best-offer.html>

Enabling Multi-jurisdiction Sales Tax

Overview

You can create a tax table for a seller for any eBay site for which tax tables are supported. (To determine if tax tables are supported for a given site ID, see “[Tax Jurisdictions](#)” on page 615.) Such tax tables specify, on a state-by-state or jurisdiction-by-jurisdiction basis, what tax would be charged for that jurisdiction and whether fees related to shipping and handling are to be part of the total that is taxed.

Sales tax is charged based on the buyer's shipping address (independent of the site from which the buyer buys) and on whether tax is to be paid for that jurisdiction. For example, if the item was listed on the US site, the seller provided tax details for Ohio and Florida but not for California, and a buyer in California specifies a shipping address in Ohio, the buyer will be charged the Ohio tax.

There is an association between a user, a tax table and a site. A tax table can be created for a user on a listing site that supports tax tables. A user can have multiple tax tables (as many tables as there are sites that have jurisdictions and support tax tables). It is the listing site of the item that determines which tax table to apply during the checkout process.

The US and eBay Motors US sites share the same global tax table: changing the table on one site changes it for the other.

After having created a tax table, a seller is offered the choice of charging no tax or of charging tax according to the seller's tax table for that site when listing items.

Jurisdiction taxes are separate from and not related to VAT taxes.

Miscellaneous tax field and tax table details:

- For a site to charge tax, it must have tax jurisdictions.
- A site may or may not support tax tables. Taxes can still be specified for a site that has jurisdictions, yet does not support tax tables: there are the single [SalesTaxState](#), [SalesTaxPercent](#), and [ShippingIncludedInTax](#) fields.
- A user can only have a tax table if the listing site has jurisdictions and supports tax tables.

Using Tax Tables

The following calls and fields enable multi-jurisdiction sales tax:

- [SetTaxTable](#)
- [GetTaxTable](#)
- The [AddItem](#) family of calls: [Item.UseTaxTable](#)
- The [GetItem](#), [GetBidderList](#), [GetSellerList](#) family of calls: [Item.ShippingDetails.TaxTable](#)
- The [GetSeller/ItemTransactions](#), [GetOrders](#) family of calls: [Transaction.ShippingDetails.TaxTable](#)

TaxJurisdiction.JurisdictionID is typically an abbreviation representing a jurisdiction. For the US site, all **JurisdictionID** values are two-letter abbreviations consistent with US postal abbreviations. (For example, the value of **JurisdictionID** for California is **CA**.) For a list of valid jurisdictions per site, see “[Tax Jurisdictions](#)” on page 615.

TaxJurisdiction.JurisdictionID must match one of the valid jurisdictions for that site, as obtained via [GetTaxTable](#). For example, **CA** is a valid jurisdiction **JurisdictionID** for the US site while **California** is not.

Whatever you send to [SetTaxTable](#) is considered to be a complete tax table. Thus, if your goal is to modify certain values in the table while preserving others, you must provide details for all. Omitting details for a jurisdiction eliminates that jurisdiction from the tax table altogether. (Note the difference from a function like [Reviseltem](#) which retains a value of a field unless overridden.)

Various functions return **TaxJurisdiction** blocks along with the older fields, **SalesTaxPercent**, **SalesTaxState**, and **ShippingIncludedInTax**. Applications should inspect the **TaxJurisdiction** blocks for tax information.

How Tax Tables Are Applied

A tax table for a seller and site is a global preference. As such, changes to it take effect immediately. However, changes to such a table do not affect active listings.

When an item is listed, the current tax table for that user and site is applied if **UseTaxTable** is true (or the single state/rate fields are used if the seller is not a tax table user). Currently-listed items are not affected by changes to the tax table or to tax table policy.

In general, when an item is revised or relisted, the following rules apply:

- If no tax was originally specified for the item, the item remains without tax.
- If tax was specified for the item via the **SalesTaxPercent**, **SalesTaxState** and **ShippingIncludedInTax** fields (in **ShippingDetails.SalesTax**):
 - if a tax table exists for the user and site, the tax table is applied.
 - if no tax table exists (or the tax table has been cleared), tax details are unchanged.
- If tax was applied via tax table:
 - if a tax table exists for the user and site, the tax table is applied.
 - if no tax table exists (or the tax table has been cleared), no taxes are charged.

One exception when revising items is that tax details cannot be changed for active listings that already have bids.

GetItem always returns the tax table originally associated with the item.

When an order line item is created, the order line item is assigned a copy of the tax table assigned to the item.

A Combined Payment order is a collection of two or more order line items between the same buyer and seller. A Combined Payment order can be created through the eBay site or through the **AddOrder**. When created through the eBay site, the order takes on the tax table of the default item, where the default item is the item with which the buyer began the checkout process (e.g. the buyer clicked Pay Now while viewing the item or the buyer selected the first item in the My eBay listing).

When created via **AddOrder**, no tax table is used; the **SalesTaxPercent**, **SalesTaxState** and **ShippingIncludedInTax** fields are the means by which taxes are specified for all sites.

If no tax information is passed via **AddOrder**, the order will not charge taxes regardless of any tax information on the individual items.

If a buyer breaks up a Combined Payment order (for example, to pay for a single order line item), the Combined Payment order is considered cancelled and the order line items roll back to their original tax tables.

With the eBay Web site, it is possible in the Send Invoice flow to specify a different jurisdiction or tax rate than was associated with the order line item (this would modify the tax table on the order line item). Thus, it is possible for a seller's tax table to be different from the tax tables of a related order line item and the Combined Payment order's tax table.

Enabling Get It Fast

Many buyers want or need to receive an item as soon as possible after purchase (for example, last-minute shopping for the holidays). Such buyers benefit from knowing if purchasing an item from one seller would get the item into the buyer's hands faster than if purchased from another seller.

Buyers can search for **Get It Fast** listings, listings that have been opted into the **Get It Fast** feature by sellers. By opting a listing into the **Get It Fast** feature, a seller commits to a one-day handling time (**DispatchTimeMax**=1, or '0' once the Same Day Handling feature is rolled out), and must offer at least one domestic expedited shipping service.

See “[Enabling Get It Fast](#)” on page 331 for more details.

Identifying Listings that Benefit Nonprofits

eBay Giving Works gives sellers a formal way to donate a percentage of their listings’ proceeds to particular nonprofit organizations. The feature also provides functionality that makes it easier for buyers to find items that are listed by nonprofit organizations or that benefit nonprofit organizations.

Sellers can list items through the normal Sell Your Item flow on the eBay site, or through the API. A seller selects a qualified nonprofit organization and specifies a percentage of the item purchase price to be used as a donation. After the item is purchased, the eBay Giving Works provider bills the seller and sends the donation amount to the nonprofit organization on the seller’s behalf. Nonprofit organizations can also list items on their own behalf through eBay Giving Works.

Note: There may be a minimum donation amount required for items listed through eBay Giving Works via the API or the Sell Your Item flow. This minimum donation amount may not be required for nonprofit charity organizations listing items on their own behalf. See the eBay online help at <http://pages.ebay.com/help/sell/selling-nonprofit.html> for specific details about this requirement.

Currently, eBay Giving Works is supported through the US, UK, and eBay Motors sites only. The Ad Format and Adult Only categories are not supported.

See the eBay online help for additional information about eBay Giving Works:

Selling to Benefit a Nonprofit Organization
<http://pages.ebay.com/help/sell/selling-nonprofit.html>

The remainder of this section covers these topics:

“[API Support for eBay Giving Works](#)” on page 234
“[Benefits of Retrieving Giving Works Information](#)” on page 236
“[Other Ways to List Charitable Items](#)” on page 236

API Support for eBay Giving Works

eBay Giving Works items are listed via the API through a partnership with the eBay Giving Works provider.

Adding a Giving Works Listing

Use these calls to add eBay Giving Works listings:

- [AddItem](#)
- [VerifyAddItem](#)

A seller must meet the following criteria to add an eBay Giving Works item.

- Be a registered charity seller on the eBay Giving Works provider site (for example, [MissionFish](#)) with an account in good standing.

Note: Beginning in May, existing registered charity sellers will no longer be considered registered charity sellers until they agree to the new Terms & Conditions and create a donation account. Sellers can re-register on the eBay site.

- Specify a valid nonprofit charity organization with [CharityID](#). For information on retrieving the [CharityID](#), see “[Retrieving the CharityID](#)” on page 235.

Note: A nonprofit charity organization can also be specified with [CharityNumber](#), but [CharityNumber](#) is being superseded by [CharityID](#).

- Specify a percentage of the listing price to donate to the nonprofit charity organization with [DonationPercent](#).

Note: There is a minimum donation amount required for items listed through eBay Giving Works via the API or the Sell Your Item flow. This minimum donation amount is not required for nonprofit charity organizations listing items on their own behalf. See the eBay online help at <http://pages.ebay.com/help/sell/selling-nonprofit.html> for specific details about this requirement.

Retrieving the CharityID

A list of valid [CharityID](#) values and their corresponding nonprofit charity organizations can be retrieved using [GetCharities](#). See “[GetCharities](#)” on page 813 for information on retrieving [CharityID](#) values with [GetCharities](#).

Retrieving a Giving Works Listing

Use these calls to retrieve eBay Giving Works listings:

- [GetItem](#)
- [GetSellerList](#)
- [GetItemTransactions](#)
- [GetSellerTransactionsGetOrderTransactions](#)

[GetItem](#), [GetSellerList](#), [GetItemTransactions](#), [GetSellerTransactions](#), and [GetOrderTransactions](#) return the name and ID of the nonprofit organization and the requested donation percentage in a [Charity](#) object.

Currently, eBay Giving Works is supported through the US, UK, and eBay Motors sites only. The Ad Format and Adult Only categories are not supported.

Revising a Listing to Add Charity

Use this call to revise an active listing:

- [ReviselItem](#)

Sellers can add a charity tag to an active listing under certain conditions. Because buyers may choose to retract a bid in an auction when a seller adds a charity, or a charitable organization may want to

review items benefitting their organization, you can only add a charity tag when there are more than 12 hours before the bidding ends.

When you revise an item, you can add a charity to a non-charity listing, but you cannot remove or change the charity designation.

The rules for adding a charity to a listing depend on the listing type:

- For an **Auction** listing, you can revise an item to add a charity if there are more than 12 hours left for the listing (whether or not the item has bids).
- For a **Fixed-Price** listing, you can revise an item to add a charity if there are more than 12 hours left for the listing, and the item has not been sold.
- You cannot revise listings to add charity for **Ad Format** listings.

Benefits of Retrieving Giving Works Information

Many buyers and sellers are dedicated to charitable giving for a variety of personal and financial reasons. Some buyers and sellers are particularly interested in donating to specific nonprofit organizations. Others just like to donate in general. A buyer may choose to do more business with nonprofit organizations, as well as sellers who give to nonprofit organizations or who support the same causes as the buyer.

Your application can benefit in several ways by retrieving eBay Giving Works information. For example:

- Integrate the donation process directly into regular business processes.
- Include information about a listing's benefiting nonprofit organization when presenting the listing to users.
- To provide more robust functionality, use `findItemsAdvanced` with a `itemFilter.name` value of `CharityOnly`. For each item returned, use `GetItem` and inspect the `Item.Charity` container in the response. With the information in this container, you can:
 - Create a report that compares the number of donations made to various charities, the donation percentages allocated to those charities, and so forth.
 - Create a report that shows all current listings that benefit a certain charity and the donation percentages allocated for each listing. This type of feature may be of particular interest to users who represent charitable organizations.
 - Create a report that lists all charities that were selected as donation recipients by eBay sellers within the last 30 days.

Other Ways to List Charitable Items

If a seller wants to donate a portion of an item's proceeds to a nonprofit organization but chooses not to use eBay Giving Works, the seller can use the item description to explain that a charitable donation will be made. However, in this case, the seller is responsible for sending the donation amount to the benefiting charity. The process is not managed through eBay Giving Works.

This also means that on the eBay Web site the item will not show up in search results for Giving Works listings, will not be promoted as a charitable listing, and will not be identified by the Giving Works icon.

Note: eBay has strict policies about listing charitable items through means other than eBay Giving Works. See the online help at <http://pages.ebay.com/help/policies/fundraising.html> for specific details on those requirements.

Similarly, a nonprofit organization can still list items without using eBay Giving Works and use the item description to explain that the purchase benefits that organization. However, in this case, it is the responsibility of the buyer to confirm that the seller represents a qualified nonprofit organization.

If a seller describes an item as a listing that benefits a nonprofit organization but does not list the item through eBay Giving Works, the **Charity** object is not applicable.

Listing Items as Private

When listing an item, the seller can optionally designate the listing as private. Doing so hides the user IDs of bidders from users viewing the active listing, except for the seller. This provides anonymity to the listing's bidders. A seller might designate a listing as private if potential buyers could be embarrassed when placing bids.

A seller can designate competitive-bidding listings and fixed-price listings as private. Sellers are advised to only use this designation when they have a specific reason to do so.

On the eBay site, the seller enables this option when defining the new item on the Sell Your Item (SYI) page. In the eBay API, this is done by passing a value of true for the **Item.PrivateListing** property when using **AddItem**. The **Private** argument is also available for use with **RelistItem** and **ReviselItem**.

When an active listing is viewed on the eBay site on the View Item (VI) page by anyone but the seller, the list of bidders is not displayed. Even for the seller, visible bidder user IDs are limited to the high bidder while the listing is still active. The seller is, of necessity, able to see the winning bidder after a competitive-bid item listing (an auction) has ended. In the API, the private designation is indicated by a value of true returned in the **Item.PrivateListing** when items are retrieved with such calls as **GetSellerList** and **GetItem**. For calls that return bidder user IDs (such as in the **Item.HighBidder** node of the **GetItem** result set, the high bidder user ID is only visible if the function requestor is the listing's seller.

When a user ID is hidden in an API call result set due to the private auction designation, the user ID field is still returned. But the value the field contains is "Invalid Request" (without the quotation marks) instead of the actual user ID.

For more information on private auctions, see the online help of the eBay site.

Private Auction (for Sellers)
<http://pages.ebay.com/help/sell/private.html>

Private Auction (for Buyers)
<http://pages.ebay.com/help/buy/private.html>

Designating a listing as private does not place limits on which eBay users may bid on the listing. This is another feature known as Pre-Approved Bidding. (For more on this feature, see the online help on the eBay site: [Pre-approve Bidders/Buyers](#). Note that the Pre-Approved Bidder feature is only available under limited circumstances.) Nor does the private designation block specific users from bidding on the listing. (For more on this, see the online help topic [Block Buyers](#) on the eB site.)

Adult Only Listings

Listing Adult Only Items

It is possible to list items that are intended for adults only. A number of restrictions and stipulations surround such items. For information on these rules, see the online help page “Adult Only category policy” on the eBay site. In addition to outlining the rules for listing adult items, this topic to some extent defines what constitutes an adult item and lists the types of adult items that cannot be listed at all (even in the adult categories).

Adult Only category policy

<http://pages.ebay.com/help/policies/adult-only.html>

When listing such items through the eBay API, the seller must be qualified to list adult items. There is no input argument that designates an adult-only item. Rather, the category in which the item is listed indicates this. An adult item must be listed to one of the Adult Only categories.

The “Adult Only category policy” page covers the rules that govern the listing of adult items. In addition to the rules covered on that reference page, note that sellers can list PayPal as a valid payment method for Adult Only items only if the seller's registration site ID and registration address are in the US. As well, buyers can use PayPal to purchase Adult Only items only if their registration site ID and registration address are in the US.

Retrieving Adult Items

eBay has specific rules and procedures regarding users viewing adult only items. For more on these rules, see the online help topic “Searching Adult Only Items” on the eBay site. This topic also discusses how eBay age-verifies a user before that user can access adult only categories.

Searching Adult Only Items

<http://pages.ebay.com/help/search/adult-only.html>

Retrieving an item from an Adult Only category requires some interactive participation on the part of the viewing user. An application cannot retrieve an adult item for a user unless that user has first gone to the eBay site and viewed an adult item. When the user goes to the eBay site to view an adult item for this first time, they have to agree to a “Terms of Use Mature Audiences Category” page. Submitting this form authorizes that user to view items in the Adult Only categories. Thereafter, an API application may make API calls (such as [GetSellerList](#) and [GetItem](#)) to retrieve adult items for that user (the user is specified as the function requestor).

The fact that a particular item is an adult item can be extrapolated from the category ID returned with the item data. However, an easier and more direct way to make this determination is by the

value returned in the `Item.ListingDetails.Adult` property. If this field returns a value of true, then it is an adult item from an Adult Only category.

Featuring Vehicles on the Mobile.de Site

The `MotorsGermanySearchable` listing upgrade is applicable to listings on the eBay Germany site only.

See “[Cross-Promoting Vehicles on the Mobile.de Site](#)” on page 572.

Using a SKU Identifier with Items

When you list an item, you can add a SKU (stock-keeping unit) to identify the item. SKUs are defined by sellers and often are used to track complex flows of products and information.

To add a SKU, use the `SKU` element of `ItemType`. (For listings on Half.com, use the `SellerInventoryID` element instead. See “[Half.com](#)” on page 574.)

If you are a seller and you define a SKU for an item, it is preferable to use the `SKU` element rather than the `ApplicationData` element. Using the `SKU` element, you can obtain the SKU you define for an item before and after an order line item is created.

A seller can use a single SKU to identify a single item or multiple items. Additionally, different sellers can use the same SKU. A SKU can be up to 50 characters long.

Using the API, you can add a SKU as follows:

- [AddItem.Item.SKU](#)
- [RelistItem.Item.SKU](#)
- [ReviselItem.Item.SKU](#)

The following also is valid: [VerifyAddItem.Item.SKU](#).

After you add a SKU, you can retrieve it as follows:

- [GetItem.Item.SKU](#)
- [GetItemTransactions.Item.SKU](#)
- [GetOrders.OrderArray.Order.TransactionArray.Transaction.Item.SKU](#)
- [GetOrderTransactions.OrderArray.Order.TransactionArray.Transaction.Variation.SKU](#)
- [GetSellerList.ItemArray.Item.SKU](#)
- [GetSellerTransactions.TransactionArray.Transaction.Item.SKU](#)

Additionally, after you add SKUs to items, you can specify the SKUs in a list (as a set of strings) to filter results. You can use two calls, as follows:

- [GetSellerTransactions.SKUArray.SKU](#)
- [GetSellerList.SKUArray.SKU](#)

In each case, specify each string in a [SKU](#) element.

Using Feature Packs to Save on Upgrade Fees

Feature packs enable you to save on fees applied to listing upgrades.

The feature packs are included in the [ListingEnhancementsCodeType](#). The feature pack names in that type do not exactly correspond to the feature pack names on the eBay site.

To use a feature pack, include the element for the feature pack when you use [AddItem](#), as in the following example:

Example 11-3 Sample XML for Using a Feature Pack

```
...
<Item>
...
<ListingEnhancement>ValuePackBundle</ListingEnhancement>
<SubTitle>This Product Works</SubTitle>
<PictureDetails>
    <GalleryType>Gallery</GalleryType>
    <GalleryURL>MyGalleryURL</GalleryURL>
    <PictureURL>MyPictureURL</PictureURL>
    <PictureURL>MyPictureURL</PictureURL>
</PictureDetails>
<ListingDesigner>
    <ThemeID>10</ThemeID>
</ListingDesigner>
...
</Item>
...
```

When you use [GetCategoryFeatures](#) (see “[GetCategoryFeatures](#)” on page 808) with no category ID and no detail level, the result set will include the following: [ValuePackEnabled](#), [ProPackPlusEnabled](#), and [ProPackEnabled](#).

This means that you can use these feature packs in some categories for listing upgrades as follows: [ValuePackBundle](#) contains the [Gallery](#), [Subtitle](#), and [ListingDesigner](#) features; [ProPackPlusBundle](#) contains the [BoldTitle](#), [Border](#), [Highlight](#), [Featured](#), and [Gallery](#) features; and [ProPackBundle](#) (available only to US and Canada eBay motor vehicle sellers) contains the [BoldTitle](#), [Border](#), [Featured](#) and [Highlight](#) features.

If you use all the elements of a feature pack (for example, [Gallery](#), [Subtitle](#), and [ListingDesigner](#)), you will be charged only the feature pack price (for example, the price for [ValuePackBundle](#)).

Additionally, if you use the **ValuePackBundle** feature pack, you should specify the sub-elements of **ListingDesigner**.

When you use the **ValuePackBundle** feature pack for an item, you must provide information for the item to be listed in the Gallery (see “[Including Pictures in the Search Results Gallery](#)” on page 213). You also must provide a Subtitle value. Although the API requires the Listing Designer to have a layout and theme that are *valid* in order to get the Value Pack price, you need not provide Listing Designer values. Note that if a seller fills out the Sell Your Item form, the seller can check Gallery, Subtitle and Listing Designer (and not choose any layout or theme) and get the Value Pack price.

When you use the **ValuePackBundle** feature pack for an item, you can add your own layout and theme (standard themes and layouts are not required).

Making Listings Available by Default on Another Site

The listing upgrade called International Site Visibility enables you to make your listing available in the default search results of another site. Also see “[Prerequisites for the International Site Visibility Listing Upgrade](#)” on page 242.

The following table describes the availability of this upgrade, based on the site on which you are listing (that is, based on the primary site):

Table 11-2 Relationships of Primary Sites to Secondary Sites

Primary Site (Site of Item Listing)	Secondary Sites Available for International Site Visibility
US (ebay.com)	eBay UK (ebay.co.uk)
Canada (ebay.ca)	eBay UK (ebay.co.uk)
UK (ebay.co.uk)	eBay North America (ebay.com and ebay.ca)
Ireland (ebay.ie)	eBay North America (ebay.com and ebay.ca)

Thus, if your primary site is the US site, the upgrade provides default visibility on the eBay UK site. And if your primary site is the Canada site, the upgrade provides default visibility on the eBay UK site. For more information, see the following:

Information for US sellers

<http://www2.ebay.com/aw/core/200803310800522.html>

Information for Canada sellers

<http://www2.ebay.com/aw/ca/200803311454332.html>

If your primary site is the UK site, the upgrade provides default visibility on the eBay North America (ebay.com and ebay.ca) sites. And if your primary site is the Ireland site, the upgrade provides default visibility on the eBay North America (ebay.com and ebay.ca) sites. For more information, see the following:

Information for UK and Ireland sellers

<http://www2.ebay.com/aw/uk/200803311620542.html>

If your primary site is the US site or Canada site, and you want a listing to be available in the default search results of the UK site, the following tag and value are used (in the **Item** container of **AddItem** or a related call):

Example 11-4 Sample XML for Visibility on the UK Site

```
...
<Item>
...
  <CrossBorderTrade>UK</CrossBorderTrade>
...
</Item>
...
```

If your primary site is the UK or Ireland site, and you want a listing to be available in the default search results of the eBay North America (ebay.com and ebay.ca) sites, the following tag and value are used (in the **Item** container of **AddItem** or a related call):

Example 11-5 Sample XML for Visibility on the US and Canada Sites

```
...
<Item>
...
  <CrossBorderTrade>North America</CrossBorderTrade>
...
</Item>
...
```

After an item is listed using the **CrossBorderTrade** field, the field is returned in the **GetItem** and **GetSellerList** calls.

To remove the **CrossBorderTrade** field using **ReviselItem** or **RelistItem**, specify a value of **None**, as follows:

Example 11-6 Sample XML for Removing the CrossBorderTrade field

```
...
<Item>
...
  <CrossBorderTrade>None</CrossBorderTrade>
...
</Item>
...
```

Prerequisites for the International Site Visibility Listing Upgrade

For a seller to list an item using the International Site Visibility feature, the item must be in a category that is valid for this feature (see “[GetCategoryFeatures](#)” on page 808).

Additionally, several other requirements must be met. For information about these requirements, see the following location:

Showing your items in search results on other eBay sites
<http://pages.ebay.com/help/sell/international-site.html>

Chapter 12

DE/AT New eBay Payment Process

Note: The new eBay payment process for the entire German and Austrian eBay marketplace is delayed indefinitely. We will keep you updated and inform you about all next steps as soon as possible. To find out more information about this delay, click the following link:
http://pages.ebay.co.uk/help/update_payment-process_background.html

The new DE/AT eBay payment process applies to all DE/AT listings. There are some eBay Germany categories (such as Motors Vehicles) exempt from the new eBay payment process. See the “[Categories Excluded from New Payment Process](#)” on page 251 section for more information on excluded categories.

Setting Payout Preferences on Site

As soon as German and Austrian sellers accept the new eBay payment process user agreement, they will be taken to eBay’s “Set up your payout options” page. Sellers have the option of receiving payouts from eBay in their PayPal or bank account. There is no fee when receiving payouts to a PayPal account. There may or may not be fees associated with payouts to a bank account. Sellers can switch this PayPal/Bank Account preference at any time.

Changes to Listing Calls in Trading API

As of February 28, 2012, all German and Austrian sellers receive a warning message when creating, revising, or relisting items if they have not accepted the new eBay payment process user agreement.

Beginning in June 2012, all German and Austrian sellers will be blocked from creating, revising, or relisting items if they have not accepted the new eBay payment process user agreement. Similarly, scheduled listings will not become active when the scheduled date is reached, nor will GTC (Good ‘Til Cancelled) fixed-price listings be relisted if this user agreement has not been accepted by the seller.

In all cases, the warning or error messages will contain a link to the new user agreement. As soon as the seller accepts the user agreement, he/she will be directed to set up eBay Payout preferences. In these preferences, the seller decides if he/she wants eBay to distribute funds to a PayPal or bank account.

Specifying Payment Methods No Longer Required

As soon as a German or Austrian seller is ramped up in the new eBay payment process, acceptable payment methods will no longer be required when creating, revising, or relisting an item. In the Add/Revise/Relist family of calls, the lone **PaymentMethods** value will default to

'StandardPayment'. The German standard payment methods are PayPal, Überweisung (EFT), and Skrill. Buyers may select any of these three payment methods when paying for the order.

If the seller attempts to add any non-standard payment methods, those payment methods are ignored.

The exception to this rule is when German or Austrian sellers are listing an item in a category that is excluded from the new eBay payment process. For excluded categories, the seller can specify 'COD', 'PayOnPickup', and/or 'Other' as payment methods in the [Item.PaymentMethods](#) field. To get a better understanding of excluded categories, see "[Categories Excluded from New Payment Process](#)" on page 251.

Changes to Order Retrieval Calls in Trading API

The next two sections discuss changes to [GetOrders](#) and other order retrieval calls to accommodate the eBay new eBay payment process.

Using GetOrders to Distinguish New Payment Process Orders

For German and Austrian new eBay payment process orders, the [PaymentMethodUsed](#) enumeration value returned in [GetOrders](#) and other order retrieval calls will always be 'StandardPayment'.

In the [GetOrders](#) and [GetOrderTransactions](#) responses, 'StandardPayment' will appear in the [OrderArray.Order.CheckoutStatus.PaymentMethod](#) field.

For [GetItemTransactions](#) and [GetSellerTransactions](#), 'StandardPayment' will appear in the [TransactionArray.Transaction.Status.PaymentMethodUsed](#) field.

The standard payment methods available to the buyers include PayPal, Überweisung (EFT), and Skrill. The exception to this rule is German and Austrian listings in eBay categories that are exempt from the new eBay payment process. See "[Categories Excluded from New Payment Process](#)" on page 251.

Retrieving eBay Seller Payout Information

Call [GetOrders](#) (or other order retrieval calls), [GetMyeBaySelling](#), or [SoldReport](#) (Merchant Data API) to retrieve seller payout information for DE/AT orders that are part of the new eBay payment process. Look for the [PaymentHoldStatus](#) field and the [PaymentHoldDetails](#) container in the response of these calls.

The [PaymentHoldStatus](#) values applicable to German and Austrian listings include 'PaymentHold', 'ReleaseConfirmed', 'ReleaseFailed', and 'ReleasePending'. The status will be 'PaymentHold' until the seller marks the order as shipped. The status will also be 'PaymentHold' if there is an eBay Buyer Protection case that the seller needs to resolve. See the [PaymentHoldStatus](#) documentation in the Trading API guide for more information on these values.

The **PaymentHoldDetails** container consists of the following information:

- **ExpectedReleaseDate** – the expected date that a seller will receive eBay's payout for the order. The seller can use Payout Details API to get more information on seller payouts.
- **NumOfReqSellerActions** – the number of possible actions that a seller can take to expedite the payout; this value should equal the number of **RequiredSellerAction** fields contained in **RequiredSellerActionArray**
- **RequiredSellerActionArray.RequiredSellerAction** field(s) – contain the possible action(s) that a seller can take to expedite the seller payout from eBay; values include 'None', 'MarkAsShipped', and 'ResolveeBPCCase'. If 'MarkAsShipped' is the value, the seller can mark an item as shipped in My eBay or through the **CompleteSale** Trading API call. If 'ResolveeBPCCase' is the value, the seller can start the process to resolve the case through the Resolution Center or by using the Resolution Case Management API.

The **ExternalTransaction** container, which contains information on the PayPal or Überweisung transaction, will no longer be returned to German and Austrian sellers. To verify if payment has been made on the order, look for the following container/fields:

Table 12-1 Checking Order/Payment Status in Order Retrieval Calls

Trading API Call(s)	Response Container(s)	Fields
GetOrders	OrderArray.Order.CheckoutStatus	<ul style="list-style-type: none"> ■ eBayPaymentStatus: look for 'NoPaymentFailure'
GetOrderTransactions		<ul style="list-style-type: none"> ■ Status: look for 'Complete' ■ PaymentMethod: will be 'StandardPayment'
GetItemTransactions	TransactionArray.Transaction.Status	<ul style="list-style-type: none"> ■ eBayPaymentStatus: look for 'NoPaymentFailure'
GetSellerTransactions		<ul style="list-style-type: none"> ■ CompleteStatus: look for 'Complete' ■ PaymentMethodUsed: will be 'StandardPayment'
GetMyeBaySelling	SoldList.OrderTransactionArray .OrderTransaction.Order.TransationArray .Transation	<ul style="list-style-type: none"> ■ SellerPaidStatus: look for 'Paid'
SoldReport	OrderDetails.CheckoutStatus	<ul style="list-style-type: none"> ■ eBayPaymentStatus: look for 'NoPaymentFailure' ■ Status: look for 'Complete' ■ PaymentMethod: will be 'StandardPayment'

Limitations of Order Update Calls

If payment for an order went through the new eBay payment process, German and Austrian sellers will be blocked from using **ReviseCheckoutStatus**. Similarly, German and Austrian sellers can not

use [CompleteSale](#) to mark an order as 'Paid' if payment for the order went through the new eBay payment process. However, they can still use [CompleteSale](#) to perform the following actions:

- Marking an item as 'Shipped' by setting the [Shipped](#) flag to 'true'. This is the most important action since sellers can expedite the new eBay payment process by marking an item as shipped as soon as a buyer pays for the order.
- Providing shipment information to the buyer, including a tracking number, using the [Shipment](#) container.
- Providing feedback to the buyer using the [FeedbackInfo](#) container.

The full functionality of the [CompleteSale](#) and [ReviseCheckoutStatus](#) calls will be available for German and Austrian listings in eBay categories that are exempt from the new eBay payment process.

New Payment Process APIs

The Payout Details API Version 1.0.0 went live with version 731. The Order Adjustment API 1.0.0 and Payout Details API Version 1.1.0 went live with version 771. These two APIs are discussed briefly in the following two sections. For full details on these services, see the corresponding documentation sets.

Payout Details API

The [Payout Details API](#) can be used by sellers listing on the German and Austrian sites to retrieve details on payouts sent (or scheduled to be sent) by eBay to the seller's PayPal or bank account. This API includes the following calls:

- [getOrderPayoutDetails](#) – upon passing in an eBay Order ID, this call retrieves detailed information on one or more seller payouts related to the eBay order.
- [getPayoutDetails](#) – upon passing in one or more payout IDs or a date range, this call retrieves detailed information on one or more seller payouts.
- [getScheduledPayouts](#) – this operation returns details for all scheduled payouts for the seller. This information includes estimated payout date for pending payouts.

The Payout Details API Version 1.1.0 includes the new [getScheduledPayouts](#) call and order line item refund data that is returned (if applicable) in all three Payout Details API calls.

Order Adjustment API

The [Order Adjustment API](#) can be used by sellers listing on the German and Austrian sites to accept or reject adjusted order amounts, as well as to issue a partial or full refund to the buyer for one or more order line items.

The Order Adjustment API includes the following calls:

- **acceptAdjustment** – upon passing in an eBay Order ID, the seller is authorizing eBay to adjust the order amount to the amount paid by the buyer, whether that amount is over or under the actual order amount. This call is only available to the seller if the buyer used EFT as the payment method.
- **rejectAdjustment** – upon passing in an eBay Order ID, the seller is rejecting the amount paid by the buyer to eBay, whether that amount is over or under the actual order amount. If the order adjustment is rejected by the seller, the buyer's payment is returned to the buyer along with an email that instructs the buyer to pay the correct amount for the order. This call is only available to the seller if the buyer used EFT as the payment method.
- **issueRefunds** – for orders going through the new eBay payment process, the seller can use this call to issue full or partial refunds for each line item in an order. In the request of this call, the seller is required to pass in the eBay Order ID, the **externalReferenceID** for the line item (returned in the **RefundArray** container of **GetOrders**), and the line item's purchase price. Optionally, the seller can pass in refund amounts for shipping and miscellaneous costs.

Unmatched EFT Payments

If the buyer uses EFT to pay for an order, eBay handles the payment matching. If eBay's matching payment is over or under the actual order amount, both the buyer and seller will be informed. The seller will be required to accept or reject the payment through My eBay or through the Order Adjustment API (See “[Order Adjustment API](#)” on page 247). If the seller accepts the payment adjustment, the order will be readjusted and all order retrieval API calls will be updated to reflect the adjustment amount. This adjustment amount will be reflected in the **AdjustmentAmount** field. If the seller rejects the payment adjustment, the buyer's payment is returned to the buyer along with a notice to pay the proper amount. If the seller takes no action to accept or reject the payment adjustment, the payment is also sent back to the buyer along with the notice to pay the correct amount.

GetOrders Changes Related to Over/Under EFT Payments

If a mismatched EFT payment occurs, the **eBayPaymentMismatchDetails** container will be returned in **GetOrders** and other order retrieval Trading API calls. The **eBayPaymentMismatchDetails** container will consist of the following fields:

- **MismatchType** – possible values are ‘OverPayment’ (indicating that buyer overpaid for the order) and ‘UnderPayment’ (indicating that buyer paid less than the order amount).
- **ActionRequiredBy** – this date/time value indicates the date that the seller must decide to accept or reject the buyer's payment amount. If no action is taken by the seller by this date, it has the same effect as rejecting the buyer's payment.

If the seller accepts the overpayment or underpayment amount, this accepted amount will be reflected in **AdjustmentAmount** field, which is returned at the order or transaction level (dependent on the order retrieval call being used).

Refunds

As soon as German and Austrian sellers are ramped up in the new eBay payment process, refunds through PayPal are no longer applicable since eBay is the recipient of the buyer's payment and not the seller. Instead, the sellers can use My eBay or the [issueRefunds](#) call of the Order Adjustment API (See "[Order Adjustment API](#)" on page 247) to issue full or partial refunds to the buyer.

A DE/AT seller can issue a refund to the buyer in the following use cases:

- A mutually agreed upon cancelled order
- Seller voluntarily issues a partial or full refund to the buyer (without an open eBay Buyer Protection case)
- Returns

The refund is issued to the buyer through the same payment method that the user selected to pay for the order (PayPal, EFT, or Skrill). If the seller's funds are insufficient to cover the amount of the refund, eBay pulls from the seller's Automatic Payment Method account to cover the deficit.

Both the buyer and seller are notified of the refund through emails and platform notifications. If the refund deposit fails, the buyer will be notified by email, and it will then be the buyer's responsibility to contact eBay Customer Support to pursue the refund. There are no fees charged to the buyer or seller during the refund process.

If eBay has already paid out the seller for an order, the seller will have to contact the buyer to arrange a refund to the buyer off of eBay.

A seller can issue a refund to a buyer up to 60 days after order creation.

Refunds Through the Site

The seller initiates the refund flow on the site by clicking on the **Issue Refund** link in My eBay, the Order Details page, the Easy Returns page, or the eBay Buyer Protection page (if an eBay Buyer Protection case is open against the order).

Refunds Through the Order Adjustment API

For new eBay payment process orders, a German or Austrian seller can issue a full or partial refund to the buyer through the [issueRefunds](#) call of the Order Adjustment API. To use this call, sellers will have to accept a refund and resolution billing agreement. The URL for this agreement will be returned as part of the [IssueRefunds](#) response.

If an eBay Buyer Protection case is open against the order, the seller must use the Resolution Center or the [issueFullRefund](#)/[issuePartialRefund](#) calls of the Resolution Case Management API to issue a refund to the buyer.

For more information on this call, see the [Order Adjustment API Users Guide](#).

GetOrders Changes Related to Refunds

The [RefundArray](#) container will be returned in [GetOrders](#) (and other order retrieval Trading API calls) if a partial or full refund has been issued to the seller. The [RefundArray](#) container will consist of one or more [Refund](#) containers, and each [Refund](#) container will consist of one or more [RefundTransactionArray](#).[RefundTransaction](#) containers (one for each order line item and refund type for which the buyer was refunded).

See the [GetOrders Call Reference](#) for more information on the [RefundArray](#) container.

New Platform Notifications

There are new platform notifications applicable to the new eBay payment process. There are two notification events related to seller payouts, and three notification events related to buyer refunds. These notification events are discussed in the next two sections.

Sellers use the [SetNotificationPreferences](#) Trading API call to subscribe to platform notifications. To subscribe to notification events, the seller uses a [NotificationEnable](#) container for each event. Below is sample XML to subscribe to the two new notification events related to seller payouts:

See the [SetNotificationPreferences Call Reference](#) for more information on subscribing to platform notification events.

If a seller would like to verify if they are subscribed to specific notification events, they can call the [GetNotificationPreferences](#) Trading API call. The seller must set the [PreferenceLevel](#) field in the [GetNotificationPreferences](#) request to 'Event'. Then look for the specific notification events in the [EventProperty.EventType](#) field of the [GetNotificationPreferences](#) request. See the [GetNotificationPreferences Call Reference](#) for more information on using this call.

ReadyToShip Notification

German and Austrian sellers and certified providers will need to subscribe to the [ReadyToShip](#) notification to be notified as soon as a buyer pays for an order. Once a seller receives this notification, the seller should ship the item as soon as possible and mark the item as 'Shipped' in My eBay or through the [CompleteSale](#) API call.

The payload of this notification is identical to the fields found in the [GetOrders](#) response. PayPal will no longer send an Instant Payment Notifications (IPN) for orders subject to the new eBay payment process.

ReadyForPayout Notification

German and Austrian sellers and certified providers will need to subscribe to the [ReadyForPayout](#) notification to be notified as soon as eBay's payout to the seller is scheduled. Once a seller receives this notification, the seller can use the [getPayoutDetails](#) or [getOrderPayoutDetails](#) calls of the Payout Details API to get more details on the payout.

The payload of this notification is identical to the fields found in the [GetOrders](#) response.

Unmatched EFT Payment Platform Notifications

German and Austrian sellers and certified providers will need to subscribe to the following new notifications to receive the latest information on unmatched EFT payments.

- **UnmatchedPaymentReceived** – this notification is sent to German and Austrian sellers when eBay determines that a buyer has paid less than or more than the order amount using the EFT payment method. Upon receiving this notification, the seller can then either accept or reject the buyer's payment amount through the site or by using the [AcceptAdjustment](#) or [RejectAdjustment](#) call of the Order Adjustment API.

The payload of this notification is identical to the fields found in the [GetOrders](#) response.

- **RefundSuccess** – this notification is sent to German and Austrian sellers when the seller's refund (using the site or the [IssueRefund](#) call of the Order Adjustment API) to the buyer is successful.

The payload of this notification is identical to the fields found in the [GetOrders](#) response.

- **RefundFailure** – this notification is sent to German and Austrian sellers when the seller's refund (using the site or the [IssueRefund](#) call of the Order Adjustment API) to the buyer fails.

The payload of this notification is identical to the fields found in the [GetOrders](#) response.

Setting the API Compatibility Level

To use the Payout Details API and to view Trading API containers/fields associated with phase 1 of the new eBay payment process implementation, sellers must upgrade their Trading API version to 731 or above. The API version number is passed in the X-EBAY-API-COMPATIBILITY-LEVEL call header.

To use the Order Adjustment API and to view Trading API containers/fields associated with phase 1 and phase 2 of the new eBay payment process implementation, sellers must upgrade their Trading API version to 771 or above. The API version number is passed in the X-EBAY-API-COMPATIBILITY-LEVEL call header.

Categories Excluded from New Payment Process

The list of excluded categories are subject to change, so sellers are encouraged to use the [GetCategoryFeatures](#) Trading API call to see if a specific category is exempt from the new eBay payment process. If [GetCategoryFeatures](#) is used, the seller should specify a category ID value in the call request, and then look for a [Category.PaymentMethod](#) value other than 'StandardPayment' in the call response. Specifically, the category is exempt from the new eBay payment process if the [Category.PaymentMethod](#) value is 'COD', 'PayOnPickup', or 'Other'.

The [GetCategoryFeatures](#) call will return the [PaymentOptionsGroup](#) field to indicate if a category is subject to the new eBay payment process. The three possible enumeration values are discussed below:

- [eBayPaymentProcessEnabled](#) – this value will be returned for categories subject to the new eBay payment process.
- [eBayPaymentProcessExcluded](#) – this value will be returned for categories exempt from the new eBay payment process.
- [NonStandardPaymentAllowed](#) – this value will be returned for categories that allow long handling times and/or down payments.

Sellers should always use [GetCategoryFeatures](#) to get the latest list of excluded categories.

Because payment of the order will occur off of eBay, the buyer cannot open an eBay Buyer Protection case for an item listed in an excluded category.

For exempt categories with long handling times and/or down payments required, the seller can specify 'COD', 'PayOnPickup', or 'Other' as accepted payment methods. With all of these payment methods, the payment is handled between the buyer and seller off of eBay.

Categories With Long Handling Time and/or Down Payments

German and Austrian listings in categories that allow long handling times and/or down payments are still subject to the new eBay payment process. In addition to the 'StandardPayment' options (PayPal, EFT, or Skrill), COD, PayOnPickup, and Other are valid payment methods (if specified by the seller).

Chapter 13

Using Description Templates

Description templates enable a seller to customize the appearance of a listing's description. Description templates are the core of the Listing Designer feature. For more options related to listing items, see “[Listing Items](#)” on page 174 and “[Describing Items in a Listing](#)” on page 206.

Prior to the introduction of this feature, sellers would embed HTML within an item's description field to improve or customize the appearance of the description. When this item was subsequently viewed on the eBay site (in a View Item page), this embedded HTML was parsed by the viewing browser as part of the View Item page's source HTML. The result was a description of the item with a much better and more appealing appearance than simple text. The Listing Designer affords much of this same capability – without requiring the seller to be familiar with HTML coding. It does this by making available a collection of stock templates that are applied to the item's description when it is viewed on the eBay site. These templates can add decorative, colorful backgrounds or borders to the item description text.

When a seller is in the process of listing an item on the eBay site, the Listing Designer is one of the (optional) steps. The seller is given the opportunity to select one Layout and one Theme. (Layouts and Themes work independently, and an item can have a Layout but no Theme or vice versa.) In the context of an API application, a list of available Layouts and Themes are retrieved using [GetDescriptionTemplates](#). Each Layout and Theme is represented by a template. The application then allows the seller to select from these Layout and Theme templates when creating a new item.

When an item is viewed on the eBay site, the selected Layout and Theme are applied to the item and affect how the description is displayed within the description rectangle of the View Item page. In API applications, the raw HTML for a Layout or Theme template is retrieved and then utilize the HTML specification to display the item's description.

In the API, associate a Theme with an item at listing time by specifying the Theme's ID (retrieved using [GetDescriptionTemplates](#)) in the `Item.ListingDesigner.ThemeID` property. Associate a Layout by specifying its ID in the `Item.ListingDesigner.LayoutID` property.

- “[Layout Templates](#)” on page 254
- “[Theme Templates](#)” on page 254
- “[Theme Groups](#)” on page 255
- “[Listing Designer Effects on the eBay API](#)” on page 256
- “[Retrieving a List of Available Listing Designer Templates](#)” on page 257
- “[Listing Items Using Listing Designer Templates](#)” on page 257
- “[Retrieving Items that Have Assigned Listing Designer Templates](#)” on page 259
- “[Revising and Relisting Items Using Listing Designer Templates](#)” on page 263

Layout Templates

The Listing Designer provides a number of pre-designed layouts for item descriptions. A Layout determines how pictures are positioned in the item description relative to the text of the description. One special Layout, called "Standard," is the equivalent of no Layout at all (the way item descriptions were displayed prior to Listing Designer). In addition to the standard Layout, a number of predefined special Layouts are available. These position pictures of the item to the left of the description text, below the text, and so on. The names of these Layout templates indicate the effects to be expected. An API application need not evaluate the textual name of a Layout template when an item is being created, but simply pass the ID for the Layout template along in the item's definition in [AddItem](#) (see "[Listing Items Using Listing Designer Templates](#)" on page 257). When displaying an item in an API application—such as after calling the [GetItem](#)—the application would use the Layout template ID returned with other item properties to decide how to position the images relative to the description text (see "[Retrieving Items that Have Assigned Listing Designer Templates](#)" on page 259).

Individual sellers can select themed templates for selling motor vehicles on the US eBay Motors site (site ID 100) and the Canadian eBay site (site ID 2). The vehicle categories in which you can use the new templates include:

- Motorcycles
- Passenger Vehicles
- Other Vehicles
- Powersports
- Parts and Accessories

The Motors listing templates are designed for listing various types of vehicles and are intended for For Sale By Owner listings. All motor vehicle templates share these characteristics:

- Support for Supersize photos.
- Support for more than 12 photos.
- Only work on browsers that support Active X.

The technique for using motor vehicle templates is the same as with standard Listing Designer templates.

For fees related to Listing Designer, follow the appropriate URL in "[Fees per Site](#)" on page 593. Fees may vary depending on site and on Motors versus individual/Stores.

Theme Templates

Listing Designer Themes are a tool with which a seller can make an item's description more colorful and appealing. Themes do this by providing eBay-supplied HTML and images for formatting the description and transposing the description text over a colorful, theme-based background image. Some Themes are based on holidays, such as Mother's Day and Valentine's Day. Others are based on

events, like the birth of a baby or the coming of Spring. Some reflect the type of item being listed, such as books and music or computers. And still others are simply decorative: circles, or a floral pattern.

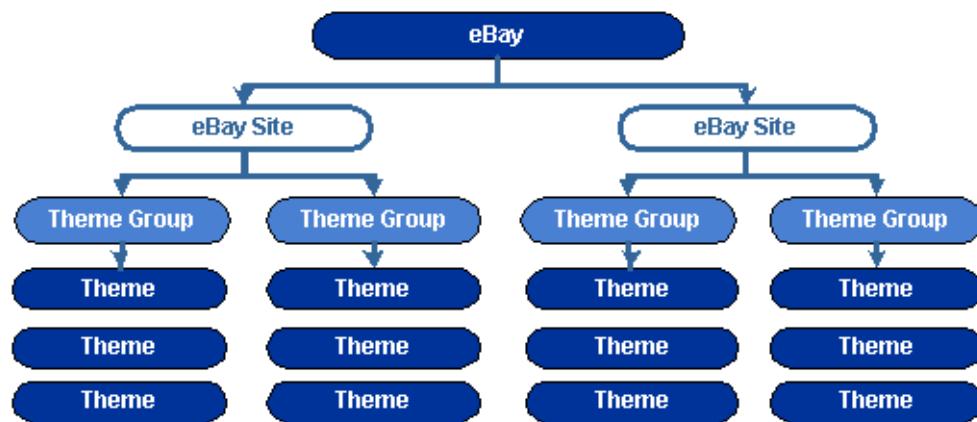
Theme templates consist the HTML for a table and a set of images. The images appear above, behind, and below the item's description, together forming an overall background for the item description display area.

When an item is first listed on the eBay site, the seller is given the opportunity to select a Theme through the Listing Designer in the eBay UI. In an API application, the Theme templates retrieved using [GetDescriptionTemplates](#) are made available to the end-user so that they can select one and assign it to the new item. When an item with an assigned Theme is displayed on the eBay site, the Theme's images appear in the background of the item description, overlaid by the description text. The images are resized horizontally as the browser window is resized. The background is also sized vertically depending on the length of the description text.

Theme Groups

Themes are segmented into logical sets, called *Theme Groups*. Theme Groups divide available Themes into intuitive categories that make it easier for a seller listing an item to find the Themes most appropriate to the seller's item, business, or listing. Data about groups are returned in the [ThemeGroup](#) nodes. API applications can use this data to store and then present this data to the seller in the same logical manner as the Themes are organized on the eBay site.

Each eBay site has its own set of Theme Groups. Each Theme Group belongs to just one site. The Theme Group name is localized to the language associated with the site it belongs to. Each Theme is a member of one Theme Group and a Theme Group may contain zero to many Themes. Theme names, too, are localized to the owning site's language.



Because of this organization, a particular Theme is only ever associated with a single eBay site. However, a Theme might be replicated across multiple sites (with a different ID, name, and owning Theme Group) to make that Theme available to sellers on different sites. For instance, a generic and very popular Theme might be present on multiple sites, though in a different naming guise on each.

On the other hand, a Theme that is specific to only a single language or culture might only appear on one site.

A call to [GetDescriptionTemplates](#) returns the Theme Groups (and thus the Themes) associated with the eBay site specified in the `Site` input argument. Each `ThemeGroup` node contains the data for one Theme Group. This data is composed of the data elements: `GroupID` (the identifier for a Theme Group), `GroupName` (the localized name of a Theme Group), `ThemelD` (a count of the Themes that are members of the Theme Group), and zero to many `ThemelD` elements (each of which is the numeric ID for one member Theme).

All of the Themes cited by ID in the various `ThemeGroup` nodes will also be returned—with their detail data—in the `DescriptionTemplate` node, elsewhere in the result set. In short, the Theme Groups provide the organization of Themes while the data returned in `DescriptionTemplate` is the detailed display data for an individual Theme template.

An application can use this information to provide the end-user with a list of Theme Groups by name. On the end-user selecting one Theme Group, the application can then display for selection just the Themes that are members of the selected Theme Group.

The Theme and Theme Group organization of data returned by [GetDescriptionTemplates](#) also lends itself to a relational database model. For instance, an application could store the Theme and Theme Group data in three related tables: `ThemeGroups` (with the IDs and names for each Theme Group), `Themes` (data for individual Themes, including the IDs), and `GroupToTheme` (which relates the other two tables through Theme Group IDs and Theme IDs). For example, to get just the Themes associated with the Theme Group that has the ID 3:

Example 13-1 Relational Database Model

```
SELECT Themes.ID, Themes.ThemeName, ...
  FROM Themes
  INNER JOIN ThemeToGroup
    ON (Themes.ID = ThemeToGroup.ThemeID)
  INNER JOIN ThemeGroups
    ON (ThemeToGroup.GroupID = ThemeGroups.ID)
   WHERE (ThemeGroups.ID = 3)
```

Lister Designer Effects on the eBay API

In addition to [GetDescriptionTemplates](#) that is solely for implementing the Listing Designer feature, other calls are also affected. The table below lists the API calls that are affected by the Listing Designer feature. The remainder of this help topic discusses in detail what those effects are.

Table 13-1 API Calls Affected by Listing Designer Feature

Call	Input	Output
AddItem	Yes	Yes
VerifyAddItem	Yes	Yes
GetItem	Yes	Yes
Reviseltem	Yes	

Table 13-1 API Calls Affected by Listing Designer Feature (Continued)

Call	Input	Output
RelistItem	Yes	

Retrieving a List of Available Listing Designer Templates

Use [GetDescriptionTemplates](#) to retrieve a list of the Listing Designer templates available on eBay. [GetDescriptionTemplates](#) returns a list of templates that includes both Layout and Theme templates. The information returned for each is shown in the table below:

Table 13-2 Information Returned, Depending on Theme versus Layout

Element	Themes	Layouts
Template category association—holidays and special events (GroupID)	Yes	
Unique identifier for a template—used to specify a template in calls (ID)	Yes	Yes
URL for the image associated with a template (ImageURL)	Yes	Yes
Textual name of a template—as appears in selection lists on the eBay site (Name)	Yes	Yes
HTML specifications for a Theme template—description display instructions (TemplateXML)	Yes	

[GetDescriptionTemplates](#) has the optional input argument [ModifyTime](#). Called without this argument, [GetDescriptionTemplates](#) returns a list of all Layout and Theme templates. But when a date is passed in [ModifyTime](#), only those templates that have been changed or added on or after the specified date are returned. When an application is enabled to use Listing Designer templates, the [GetDescriptionTemplates](#) should be called without the [ModifyTime](#) argument to compile a complete list of templates. This list should be stored locally and serve as the baseline collection of templates from which the application can select. After that, the application should periodically call [GetDescriptionTemplates](#) with the [ModifyTime](#) argument to retrieve only changed or added templates and keep this cached list up to date. Also, the [ObsoleteLayoutIDs](#) and [ObsoleteThemeIDs](#) nodes list obsolete Layout and Theme templates that should be deleted from the application's cached list.

Listing Items Using Listing Designer Templates

When an item is first listed using [AddItem](#), a Layout template or a Theme template (or both) can be assigned to the item. A Layout template is assigned to a new item by specifying the Layout template ID in the [AddItem](#) input argument [LayoutID](#). Similarly, a Theme template is assigned to the item using the [ThemelD](#) argument. Layout and Theme IDs are retrieved from eBay using [GetDescriptionTemplates](#) (see “[Retrieving a List of Available Listing Designer Templates](#)” on page 257).

Note: An application need not (and cannot) specify the Theme Group ID for the Theme specified in [ThemelD](#). A Theme ID is unique for a given eBay site and the Theme Group is inferred from the specified Theme ID. (For more on Theme Groups, see “[Theme Groups](#)” on page 255.)

In the context of an API application UI, available templates could be made available to the end-user through a multi-item selection list (e.g., a listbox). One list should be used to present the available Layout templates and another the available Theme templates. These lists would be populated with the **DescriptionTemplate.Name** values. The application would then determine the ID that corresponds to the selected name, and pass that as the value for the **LayoutID** or **ThemelD** input argument. When providing the means for a seller to select a Theme when listing a new item, the application can make use of Theme Groups to display for selection only a subset of related Themes. This makes the process easier for the seller than having to scroll through all available Themes to find the one the seller wants.

A small image (a 100x120 pixel GIF) is returned in the **DescriptionTemplate.ImageURL** field. This image provide a sample of what a Theme or Layout looks like. For Themes, this image depicts what the combined images for the Theme look like. For Layouts, the sample visually demonstrates the relative positioning of item description text and images. These images can be used in an API application along with a list of Theme and Layout names to allow the end-user to make the most informed decision on which Theme and Layout to use for an item.

The three examples of Theme samples below are for the Themes named "Tickets", "Toys and Dolls," and "Entertainment."



The three examples below are for the Layout templates named "Photo on the left," "Photo on the right," and "Photo on the bottom." They show in very visual terms how the images for the item description will be positioned relative to the description text.



Use of templates associated with the Listing Designer feature incurs a fee. The result set produced by **Addltem** returns this fee in the result set in the **ListingDesignerFee** element.

When verifying the veracity of a new item definition with **VerifyAddltem**, a test can also be made of the assignment of Listing Designer templates to the item. As with **Addltem**, a Layout template is assigned with the **LayoutID** argument and a Theme template with **ThemelD**. And, just as with **Addltem** the fee associated with use of a template is returned in the **ListingDesignerFee** element of the call's result set.

Retrieving Items that Have Assigned Listing Designer Templates

The data returned by [GetItem](#) for an item includes which Layout template and which Theme template is assigned for the item. The ID of the Layout template (if one is used) is returned in the [LayoutID](#) field and the ID for the Theme template (if any) in the [ThemID](#) field. These IDs correspond to the values returned in the [ID](#) field returned by [GetDescriptionTemplates](#) and are used to determine the particular template to use when displaying the item's description.

An API application can accommodate Listing Designer template specifications in two very different ways. The first is to have the template applied automatically. This is done by passing a true value in the [GetItem.DescFormat](#) input argument. In this case, the template HTML is returned in the [GetItem.Item.Description](#) field. If the application is displaying the item using HTML, this will work fairly easily. However, if the application does not use HTML to display the item—as might be the case for a Windows application—it will not work.

The other approach is to not have the templates automatically applied (i.e., a false value is passed in [GetItem.DescFormat](#)). In this case, the item description is returned in the [GetItem.Item.Description](#) field as simple text (with embedded HTML if the seller did that, which is totally independent of the Listing Designer feature). Under these conditions, the application must:

- Inspect [Item.ListingDesigner.LayoutID](#) and [Item.ListingDesigner.ThemID](#) return fields to identify the Layout and Theme templates.
- Retrieve the Layout and Theme specifications (from a local cache or the result set produced by [GetDescriptionTemplates](#)).
- Manually use the Layout and Theme template specifications to properly display the item's description as the seller intended.

For applications that use HTML to display an item, the template specifications must be programmatically wrapped around the raw text of the description to produce well-formed HTML. For non-HTML applications, this approach is virtually required. Such applications would need to extrapolate what non-HTML UI tools would be needed to replicate the HTML of the items associated templates. For example, an HTML application uses a simple `` tag to display an image associated with a Theme, while a non-HTML application would need to use a graphics function or UI object to do the same thing (a Windows program might use something like the Windows API function `BitBlt()` to draw the image in the appropriate place on the screen).

Displaying Item Descriptions with Layout Templates

Layout templates determine the relative positioning of the text and images within the rectangular area in which an item's description is displayed. For example, one Layout specifies that the image of the item appears in the description to the left of the description text (Layout ID 10002, name "Photo on the left"). Other Layout templates specify different relational positioning of the image and text within the rectangular area in which the description is displayed (photo on the right, photo on the bottom, etc.).

HTML-based applications would compose the HTML for the image and description text to place the image relative to the text per the Layout template. For example, if the Layout template specifies that

the picture appear to the left of the text, this could be accomplished by using the ALIGN attribute of the tag:

Example 13-2 IMG 'align' example

```

<p>The item's textual description...</p>
```

Non-HTML applications would need to programmatically draw the image and the text in relative positions in the description display area that are in compliance with the Layout specification (image to the left, text to the right).

Displaying Item Descriptions with Theme Templates

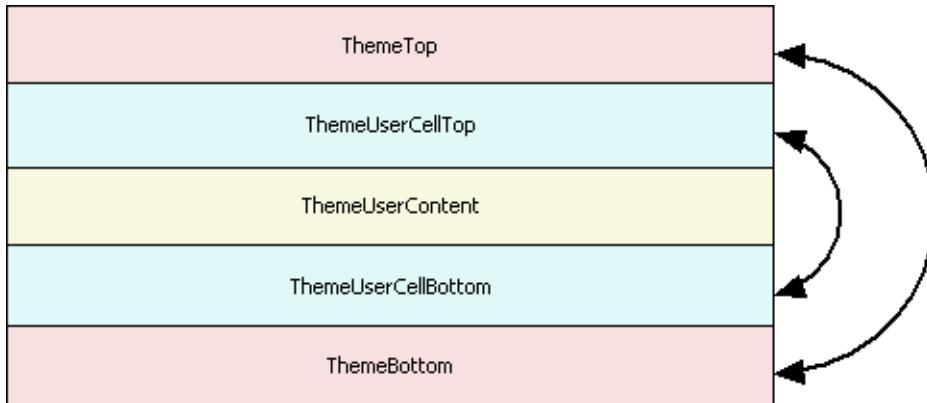
Theme template specifications are returned in five [GetDescriptionTemplates](#) return fields. Each of these fields supplies the HTML code that affects one of five sectors within the rectangular area in which the item description is displayed. Each of these return fields contains one part of an HTML <TABLE> construct that, when taken together and in the proper order are combined to the well-formed HTML for a table. Each might include image references (depending on the particular Theme), which are used for the background of one table cell (using the BACKGROUND attribute of the HTML tag <TD>).

The table below lists the Theme template return fields, indicating the sector each affects and what the effects are. They are listed here in the order of the vertical sectors of the description display area.

Table 13-3 Return fields for Theme template

Field	Sector
ThemeTop	Appears at the very top of the description display, above the seller's description content. Contains the beginning of the HTML table. May contain an image, color specification, or text that would appear at the top of the description and above any text the seller specified.
ThemeUserCellTop	Typically non-visual (i.e., no image references), the HTML returned in this field has the HTML that formats the seller's description text. Contains the beginning of the <TR> and <TD> construct that surrounds the text and photo supplied by the seller. Will typically have such text formatting as text alignment (ALIGN attribute of <TD>) and background color for the description text (BGCOLOR attribute of <TD>).
ThemeUserContent	Insertions into the description (if any).
ThemeUserCellBottom	Typically non-visual, the HTML returned in this field has the HTML that formats the seller's description text. Contains the end of the <TR> and <TD> construct that surrounds the text and photo supplied by the seller. (Ends the HTML begun in ThemeUserCellTop.)
ThemeBottom	Appears at the very bottom of the description display, below the seller's description content. Contains the end of the HTML table. May contain an image, color specification, or text that would appear at the bottom of the description and below any text the seller specified. (Completes the HTML begun in ThemeTop.)

The diagram below visually illustrates the sectors (or *bands*) within the description display area that each of these Theme template fields affects.



One example of a Theme template that has theme-based images both above and below the seller's description content is the one entitled "Boutique." The HTML specifications for the elements of this template are shown in the table below. Note that all of the bands except **ThemeUserContent** are shown as they are returned by **GetDescriptionTemplates**. This particular theme does not return an HTML specification for this band in the **ThemeUserContent** node of the **GetDescriptionTemplates** result set. The seller's description is all that is specified for that band.

Table 13-4 Sample: HTML Specifications of the 'Boutique' template

Field	HTML
ThemeTop	<TABLE CELLPADDING="0" CELLSPACING="0" BORDER="0" WIDTH="100%">><TR><TD BGCOLOR="#9999CC" WIDTH="2%"></TD><TD BACKGROUND="http://pics.ebay.com/aw/pics/vit/boutique/BoutiqueTop_40x48.gif" ALIGN="center" WIDTH="96%"></TD><TD BGCOLOR="#9999CC" WIDTH="2%"></TD></TR>
ThemeUserCellTop	<TR><TD BGCOLOR="#9999CC"></TD><TD ALIGN="center">
ThemeUserContent	Seller's item description text appears here...
ThemeUserCellBottom	</TD><TD BGCOLOR="#9999CC"></TD></TR>
ThemeBottom	<TR><TD BGCOLOR="#9999CC"></TD><TD BACKGROUND="http://pics.ebay.com/aw/pics/vit/boutique/BoutiqueBot_40x48.gif" ALIGN="center"></TD><TD BGCOLOR="#9999CC"></TD></TR></TABLE>

As can be seen, the **ThemeTop** value includes opening of the HTML <TABLE> construct and includes multiple images that appear in its band, above the seller's description content. The multiple images are:

http://pics.ebay.com/aw/pics/vit/boutique/BoutiqueTop_40x48.gif (background for the <TD> band above the seller's description);



http://pics.ebay.com/aw/pics/vit/boutique/BoutiqueCtrTop_240x48.gif (displayed in the <TD> band above the seller's description, superimposed over the background image):



http://pics.ebay.com/aw/pics/vit/boutique/BoutiqueBot_40x48.gif (background image for the <TD> band below the seller's description):



http://pics.ebay.com/aw/pics/vit/boutique/BoutiqueCtrBot_240x48.gif (displayed in the <TD> band below the seller's description, superimposed over the background image):



The **ThemeUserCellTop** has the start of the <TD> construct that surrounds the seller's description content, sets the background color behind the content, and centers the description content. The **ThemeUserCellBottom** value simply closes out the <TD> construct that was begun in **ThemeUserCellTop**, finishing the wrapper around the seller's description content. Finally, the **ThemeBottom** value includes an image that appears below the description content and closes out the <TABLE> construct that was started in **ThemeTop**, completing the template. (The image x.gif that appears here is an invisible image used simply for spacing purposes.)

Put together, these four elements of the Theme template for Tickets (**ThemeTop**, **ThemeUserCellTop**, **ThemeUserCellBottom**, and **ThemeBottom**) combined with some item description text would look like this:



An HTML-based application that is handling the template elements manually would retrieve the elements (from a local cache or using [GetDescriptionTemplates](#)) and combine them together along with the item description text to form the complete <TABLE> construct. This complete construct would then be included in the HTML page used to display the item.

A non-HTML application would need to retrieve the individual template elements, parse the HTML them, and then translate the table elements and image references into UI elements particular to the programming language used.

Revising and Relisting Items Using Listing Designer Templates

When an item's definition is changed using [ReviselItem](#), associations of Listing Designer templates with the item can be created, changed, or removed. To assign a Layout template to the item (or change the assignment to a different template), pass the ID of the template in the [LayoutID](#) input argument. Similarly, to assign a Theme template to the item, pass the template ID in the [ThemelID](#) argument. Valid Listing Designer template ID are obtained using a call to [GetDescriptionTemplates](#) (see [Retrieving a List of Available Listing Designer Templates](#)). To remove a Layout template assignment, pass a value of 10000 (for the "Standard" Layout) in the [LayoutID](#) argument. To remove a Theme assignment, pass the Theme ID for the template named "None" (which for the US is 10) in the [ThemelID](#) argument.

Note: The Theme ID for the "None" template varies from one eBay site to the next. The actual ID can be derived by concatenating the site ID and the literal "10" and using the result as an integer ID. For example, the site ID for the US eBay site is 0. So the Theme ID for "None" is 10 ("0" plus "10"). But for the UK site, the ID is 310 (site ID "3" plus "10").

As with all for-fee features, there are restrictions on when the Listing Designer template assignments can and cannot be changed. Basically, if at least one bid has been placed on the item [or](#) there are less than 12 hours remaining in the auction, the template settings for the item cannot be changed (using [ReviselItem](#) or the eBay site).

[RelistItem](#)—used to relist an item—also supports Listing Designer templates. When the item is being relisted, assign a Layout template by passing the appropriate template ID in the [LayoutID](#) argument. To assign a Theme template to the item, pass the ID for the template in the [ThemelID](#) argument.

Chapter 14

Multi-Variation Listings

Multi-variation listings contain multiple items that are essentially the same product, but that vary in their manufacturing details or packaging.

Figure 14-1 T-Shirt Variations



For example, a particular brand and style of shirt could be available in different sizes and colors, such as “large blue” and “medium black”.

The eBay Website online help provides additional information:

Multi-Quantity Fixed Price Listings with Variations (US)

<http://pages.ebay.com/sellerinformation/sellingresources/multiplevariations.html>

This feature depends on using custom item specifics and multi-quantity fixed-price listing formats. See “[Working with Custom Item Specifics](#)” on page 481 and “[Listing Types \(Formats\)](#)” on page 136.

Variations are supported across many areas of eBay, such as listing, buying, My eBay, search, and order management.

Note: `AddItem`, `ReviselItem`, `RelistItem`, and `VerifyAddItem` do not support multi-variation listings. Instead, you use the fixed-price versions of these calls, which also support inventory tracking by the seller’s SKU: `AddFixedPricelItem`, `ReviseFixedPricelItem`, `RelistFixedPricelItem`, and `VerifyFixedPricelItem`.

This chapter covers these topics:

- “[Overview of Multi-Variation Listings](#)” on page 265
- “[Qualifications for Listing with Variations](#)” on page 267
- “[Describing Variations in a Listing](#)” on page 268
- “[Revising and Relisting with Variations](#)” on page 277
- “[Retrieving a Listing’s Variations](#)” on page 280
- “[Buying Items in Multi-Variation Listings](#)” on page 280
- “[Retrieving Order Line Items with Variations](#)” on page 281
- “[Identifying Variations in Member Messages](#)” on page 281

Overview of Multi-Variation Listings

Multi-variation listings give you a structured way to differentiate between similar (but not identical) items in a multi-quantity fixed-price listing. For example:

- A single listing that sells women's shirts could contain multiple items of the same brand and style that vary by size and color (like "Large, Blue" and "Medium, Black").
- A single listing that sells window treatments (curtains) could contain multiple items of the same brand that vary by color, length, and width.
- A single listing that sells golf clubs could contain multiple items of the same brand and model that vary by dexterity, shaft material, shaft flex, and loft.

Figure 14-2 shows an example of eBay's View Item page with a shirt variations.

Figure 14-2 Multi-Variation Listing (Buyer View)

eBay offers this feature in categories where sellers tend to list very similar items, such as Clothing, Shoes & Accessories. Check [VariationsEnabled](#) in [GetCategoryFeatures](#) to determine which categories support multi-variation listings. A summary of supported categories is also available in the eBay online help.

What Categories Can Have Variations? (US)

<http://pages.ebay.com/help/sell/listing-variations.html#categories>

What Categories Can Have Variations? (UK)

<http://pages.ebay.co.uk/help/sell/listing-variations.html#categories>

Bei welchen Kategorien können Sie Variante einstellen? (DE)

<http://pages.ebay.de/help/sell/listing-variations.html#categories>

Figure 14-3 shows one way an application could present sellers with options to include variations in a single listing.

Figure 14-3 Multi-Variation Listing (Seller View)

The screenshot shows a eBay seller interface for listing variations. At the top, there are tabs: 'LIST YOUR VARIATIONS' (highlighted), 'PICTURES', and 'QUANTITY AND PRICE'. Below the tabs, it says 'Enter quantity and price' and 'Enter quantity and price for the variations you're selling'. There is a 'Save for later' button and a 'Help' link. The main area is a table with columns: Picture, Custom label, Size type, Size, Color, Quantity, and Fixed price. The table contains 12 rows, each representing a variation of a t-shirt. The variations are categorized by color (Pink, Black, Blue, Champagne) and size type (Petites). Each row includes a checkbox, a picture of the t-shirt, the variation label (e.g., TShirt_Pk_PS), size type (Petites), size (PS, PM, PL), color (Pink, Black, Blue, Champagne), quantity (10 items), and fixed price (\$15).

Enter quantity and price for the variations you're selling						
<input type="checkbox"/>	Edit fixed price	Remove				Preview your listing
Picture	Custom label	Size type	Size	Color	Quantity	Fixed price
<input type="checkbox"/>	TShirt_Pk_PS	Petites	PS	Pink	10 items	\$ 15
<input type="checkbox"/>	TShirt_Pk_PM	Petites	PM	Pink	8 items	\$ 15
<input type="checkbox"/>	TShirt_Pk_PL	Petites	PL	Pink	5 items	\$ 15
<input type="checkbox"/>	TShirt_Blk_PS	Petites	PS	Black	10 items	\$ 15
<input type="checkbox"/>	TShirt_Blk_PM	Petites	PM	Black	10 items	\$ 15
<input type="checkbox"/>	TShirt_Blk_PL	Petites	PL	Black	10 items	\$ 15
<input type="checkbox"/>	TShirt_Ble_PS	Petites	PS	Blue	10 items	\$ 15
<input type="checkbox"/>	TShirt_Ble_PM	Petites	PM	Blue	10 items	\$ 15
<input type="checkbox"/>	TShirt_Ble_PL	Petites	PL	Blue	10 items	\$ 15
<input type="checkbox"/>	TShirt_Chgne_PS	Petites	PS	Champagne	10 items	\$ 15
<input type="checkbox"/>	TShirt_Chgne_PM	Petites	PM	Champagne	10 items	\$ 15
<input type="checkbox"/>	TShirt_Chgne_PL	Petites	PL	Champagne	10 items	\$ 15

[Edit fixed price](#) [Remove](#) [Preview your listing](#)

eBay's multi-variation listing feature provides several benefits for sellers:

- Instead of separately listing similar items with different quantities (such as 10 small, 8 medium, and 5 large pink shirts), all the items can be consolidated into a single listing.
 - Each variation can have its own quantity, price, and an optional SKU.
 - The listing can specify shared custom item specifics (such as Brand), and each variation can also have its own variation-level custom item specifics (also referred to as "variation details" on the eBay Web site) that distinguish each variation in View Item and in search results.
- There is no additional fee for listing with variations. The fee is the same as the standard fee for any fixed-price listing. In fact, as you can create one listing instead of creating multiple similar listings, you are likely to experience an overall reduction in insertion fees.
- In addition to the standard set of pictures for the listing, you can include pictures for each variation.
 - The total picture fees, including upgrades such as Picture Pack, are based on the largest set of pictures in the listing. This could be the standard (shared) set of pictures or a set of pictures associated with one of the variations. The set with the most pictures determines the fees.
 - Smaller sets of pictures (associated with the other variations in the listing) incur no additional fees.

Selling multiple variations from the same listing is the same as selling multiple items from any other fixed-price listing.

Note: For Best Match information related to multi-variation listings, see the Best Match information at the following topic:
[Multi-quantity Fixed Price listings with variations](#).

Qualifications for Listing with Variations

For a listing with variations to be accepted by eBay, the seller and the listing must meet some basic qualifications.

See the eBay online help for details about item and seller qualifications. If you search for "variations" in your site's language, you will find help links similar to these:

Creating a Listing with Variations (US)
<http://pages.ebay.com/help/sell/listing-variations.html>

Listing Items with Variations (UK)
<http://pages.ebay.co.uk/help/sell/listing-variations.html>

Angebote mit mehreren Varianten einstellen (DE)
<http://pages.ebay.de/help/sell/listing-variations.html>

If a seller becomes ineligible to create multi-variation listings, they can't create new listings with variations. However, their existing listings with variations can continue (and be revised) until the

next “Good ‘Til Cancelled” (GTC) renewal or until ended by other means (such as items sold out or end time reached).

In the rare case that a category stops supporting multi-variation listings, existing listings with variations can continue (and be revised) until their next GTC renewal or until ended by other means (such as items sold out or end time reached).

Describing Variations in a Listing

To create and modify listings with variations, you use Add/Revise/Relist family of API calls. Except where specified in this documentation, listings with variations follow the same rules and support the same functionality as any other fixed-price listing.

These are the main components that you need to define for multi-variation listings:

- The listing’s variations, with item specifics, quantity, price, and (optionally) SKU (see “[Defining a Variation](#)” on page 269)
- Instructions for validating and presenting variation-level item specifics (variation specifics) in buying tools, such as the View Item page (see “[Configuring Variation-Selection Widgets for Buyers \(View Item\)](#)” on page 273)
- Pictures that will be associated with the variations in View Item and in search results (see “[Defining Pictures for Variations](#)” on page 275)

Also note the following requirements and recommendations:

- The maximum number of variations allowed for any listing is 250.

This value is subject to change. If your application retrieves listings (using `GetItem` and related calls), we suggest you design it to accommodate listings that contain more than 250 variations.

- Each variation in a listing can have up to five variation details (in other words, up to five `Item.Variations.Variation.VariationSpecifics` nodes in the `AddFixedPriceItem` call), and up to 30 values for each variation details (in other words, up to 30 `Item.Variations.Variation.VariationSpecifics.NameValueList.Value` values under each `VariationSpecifics` node).

To give an example, for a multi-variation shoes listing, the variation details could include Color, Size, Width, Material, and Style, and the Color values could include Blue, White, Green, and Black.

- The recommended number of variations is 2 or more.

To support cases where the seller may have a business need to list an item and then add more variations to it later (when more inventory becomes available), the minimum requirement is 1 variation per listing. However, we recommend a minimum of 2 variations when you list with this feature. When buyers see a multi-variation listing in eBay’s View Item page, they’ll be offered drop-down lists from which to choose variations. eBay’s research has shown that buyers are confused when only 1 variation is offered.

You can partially control the order in which variation nodes are presented:

- The order in which you specify the **Variation** nodes in **AddFixedPricelItem** or **RelistFixedPricelItem** is the order in which they appear in View Item, and the order in which they're returned in **GetItem**.
- If you subsequently add a new variation when you revise a listing, the new variation appears at the *end* of the list (regardless of where you specify it in **ReviseFixedPricelItem**).
- You can control the order of the variation-selection drop-downs that are presented in eBay's View Item page (see "[Configuring Variation-Selection Widgets for Buyers \(View Item\)](#)" on page 273).

Defining a Variation

Use **VariationsEnabled** in [GetCategoryFeatures](#) to determine which categories support multi-variation listings.

In the **AddFixedPricelItem** family of calls, use the **Item.Variations.Variation** node to describe each variation.

Each variation should provide enough detail to distinguish the variation from other variations in the same listing. For example, if the items vary by color and size, each variation would specify one of those colors and sizes.

For details and rules, see the following sections:

- ["Setting Item Specifics for Variations" on page 269](#)
- ["Using eBay Recommendations for Variation Specifics" on page 271](#)
- ["Setting the Variation Price" on page 271](#)
- ["Setting and Modifying a Variation's Quantity" on page 272](#)
- ["Setting a SKU to Uniquely Identify the Variation" on page 272](#)

Setting Item Specifics for Variations

Item specifics are name/value pairs (or a name with a list of values, in some cases) that give you a structured way to describe common details about items.

- Use **Item.ItemSpecifics** to define item specifics that don't differ, such as Brand.
- Use **Item.Variations.VariationSpecifics** for details that differentiate each variation.

To help distinguish the item-level and variation-level details, this documentation (and the API) uses the term *variation specifics* for the variation-level item specifics. See the [XML sample](#) provided in the **AddFixedPricelItem** call reference for an example.

Note: This section focuses on how to specify variation specifics for individual variations.

For your listing request to succeed, you also need to specify the full set of variation specifics that buyers can choose from. See "[Configuring Variation-Selection Widgets for Buyers \(View Item\)](#)" on page 273.

The set of variation specifics within a variation needs to uniquely identify that variation within the listing. Therefore, the variation specifics for each variation must differ by at least one value. For example, a listing could have the following variations, where each combination of names and values is unique:

- Color=Blue, Size=M
- Color=Blue, Size=L
- Color=Black, Size=M

You can include up to five (5) name/value pairs in each variation.

For information on names that might be required in some categories, and recommendations for names that buyers are likely to search for, see ["Using eBay Recommendations for Variation Specifics"](#) on page 271.

If you duplicate a combination of names and values in the same listing, your listing request will fail. Therefore, your application should check for uniqueness across the variations before you submit a listing.

All variations within a listing need to specify the same set of names. For example, suppose you're listing petite and standard-sized T-shirts that have a stripe across the chest, and they're available with short or long sleeves. For each variation, you might use these five names: Size Type, Size, Color, Stripe Color, and Sleeve Style. Every variation in the listing needs to have the same number of names. That is, you can't specify one variation with five names and another variation with only three of those names.

The names in the variation specifics can't match any names in the shared item specifics. In the variation specifics, only use names that help the buyer distinguish the variations from each other.

Call [GeteBayDetails](#) to check the latest maximum characters for names and values in custom item specifics (typically 70 or less).

Each name can have up to 30 different values defined across all the variations in a listing. For example, you could have shirts in 30 different colors.

Note: As described earlier in this document, a listing can include a maximum of 250 variations. You may need to consider the maximum allowed of names and values to determine how to distribute your inventory.

For example, suppose you have shirts in 20 different colors. You offer 5 sizes (S, M, L, XL, XXL). For each size, you offer 2 sleeve styles (e.g., Short Sleeved and Long Sleeved).

In this case, the initial total number of variations could be 200 (20 colors x 5 sizes x 2 sleeve styles).

Then you decide to also add petite sizes for each color (PS, PM, PL).

Now the total number of variations could be 320 (20 colors x 8 sizes x 2 sleeve styles)--which is too many variations for one listing.

To solve this, you could create separate listings for the sleeve styles. In this case, Sleeve Style would become a shared Item Specific, and the total number of variations in each listing would be 160 (20 colors x 8 sizes).

Using eBay Recommendations for Variation Specifics

Call [GetCategorySpecifics](#) to determine names that most sellers use and/or that most buyers search for. Some category-specific rules are also returned. For example:

- Most categories support free-text names and values in custom item specifics and in variations. However, some categories require certain item specifics (e.g., Brand), and some item specifics (such as Color) may require you to select a value from a predefined set.

If an item specific name is required, you can either specify it in the item-level item specifics (if it is applicable to all the variations) or within the variations (if its value changes across the variations). For example, you might specify Brand at the item level and Color within the variations.

- Some categories return a [VariationSpecifics](#) flag that identifies names (such as Color) that are recommended as variation specifics, and/or a [VariationPicture](#) flag that identifies names that are recommended for variation pictures (see “[Defining Pictures for Variations](#)” on page 275).
- If the recommendations say that only 1 value is allowed in item specifics, you can still specify multiple values (one per variation) in the variations. For example, if [GetCategorySpecifics](#) says that Size Type allows 1 value ([MaxValues=1](#)), you have two choices:
 - Specify Size Type in the item specifics, with 1 value
 - Specify Size Type in the variation specifics, with 1 value for each variation
- Like other item specifics, variation specifics can return [Relationship](#) nodes that define parent/child relationships, such as Size Type (Regular, Petite, etc.) and Size (Small, Medium, Large).

You can improve your item’s chances of being surfaced in search results by specifying both the parent and the child variations in your listing when possible. For example, if you specify Size but not Size Type, and then a buyer searches against Size Type, your item might not be surfaced as a high match.

If your application has a graphical user interface, grouping parent and child fields together can help sellers understand the relationships.

Figure 14-4 Variation Relationships



Setting the Variation Price

In [AddFixedPricelItem](#), specify the price of each variation in [Variation.StartPrice](#).

The rules for variation prices are the same for the rules for any fixed-price listing. The price of each variation must satisfy the minimum price requirement for the site and category of the listing. The

currency must match the listing site's currency. If a site or category has minimum price requirements for cross-border trade visibility, the same minimums would apply to the variation prices.

When you list with variations, do not specify [Item.StartPrice](#). (If you do, it will be dropped with a warning.)

In [GetItem](#) and related calls, eBay sets the item-level prices (like [Item.StartPrice](#), [Item.SellingStatus.CurrentPrice](#), and other prices) based on the lowest-price variation that is available for purchase (with a price and quantity greater than zero). This is necessary for backward-compatibility (to support applications that are not yet variation-aware).

Note: The listing start price is recalculated within 15 minutes of an applicable event, such as a price change or when the quantity of the lowest-price variation reaches zero.

You can revise the price of a variation as long as the quantity is greater than zero (or if you increase the quantity to a non-zero value at the same time), even after it has purchases. This enables you to adjust the price competitively and to offer discounts or sales as needed.

Setting and Modifying a Variation's Quantity

Specify each variation's item quantity in [Variation.Quantity](#). This must be a non-zero value when the variation is created.

When you list with variations, do not specify [Item.Quantity](#). (If you do, it will be dropped with a warning.)

In [GetItem](#), eBay shows the overall listing quantity as the sum (total) of the quantities of all the variations in the listing. This total quantity is also used to calculate any quantity-based listing fees or credits.

When items from a variation sell, the [Variation.Quantity](#) retains its original value, and [Variation.SellingStatus.QuantitySold](#) is updated (in [GetItem](#) and related calls). To determine the remaining quantity available, subtract [QuantitySold](#) from [Quantity](#).

Listings that include variations don't support lots ([Item.LotSize](#)).

For information about changing quantities, see:

- “[Modifying a Variation's Quantity when Revising](#)” on page 279
- “[Modifying a Variation's Quantity When Relisting](#)” on page 280

Setting a SKU to Uniquely Identify the Variation

SKUs are defined by sellers and often are used to track complex flows of products and information. When you create a fixed-price listing, you can include a SKU for your own reference. You can also specify a SKU for each variation.

If the item-level SKU is unique across all of your active eBay listings, you can use it as the listing's identifier. This means you can pass your SKU instead of the item ID in subsequent requests, such as [ReviseFixedPriceltem](#). However, you must make this choice when you first create the listing.

When you create a listing with [AddFixedPricelItem](#) or [RelistFixedPricelItem](#), use the [InventoryTrackingMethod](#) field to choose whether you prefer to identify the listing by item ID or by SKU. (In either case, you still have the ability to use the item ID as needed.)

You can also specify a SKU for each variation. The variation-level SKU values must be unique within the listing (and can't be the same as the [Item.SKU](#) value, if any).

When you choose to track a listing (and variations) by SKU, you can't create and track new listings by the same SKU while the existing listing is active (that is, until the existing listing with that SKU has ended). In other words, the [Item.SKU](#) value must be unique across all of your active listings that have [Item.InventoryTrackingMethod](#) set to SKU. (SKU does not need to be unique across listings that are tracked by item ID.)

When you track by item IDs, the variation-level SKU values are optional. However, if [InventoryTrackingMethod](#) is set to SKU, then the variation-level SKU values are required, and they must be unique across all the seller's active listings. (You can reuse the SKU for a new listing or variation after the listing that previously used it has ended.)

If [InventoryTrackingMethod](#) is set to SKU, you cannot modify variation-level SKU values when you revise or relist.

Note: The eBay Web site UI might not uniquely identify listings by SKU. For example, My eBay pages and Search pages all identify listings by item ID. When a buyer contacts you via eBay's messaging functionality, eBay uses the item ID as the identifier. Buyer-focused APIs (like the Shopping API) also do not support SKU as an identifier.

The [AddItem](#) and [RelistItem](#) calls do not offer the inventory tracking preference. Only [AddFixedPricelItem](#) and [RelistFixedPricelItem](#) offer this preference.

Optionally, you can retrieve order line items for specific SKUs by specifying [SKUArray](#) in [GetSellerTransactions](#).

Note: [GetItemTransactions](#) doesn't support SKU as an input because [GetItemTransactions](#) requires an identifier that is unique across your active and ended listings, such as [OrderLineItemID](#). Even when [InventoryTrackingMethod](#) is set to SKU in a listing, eBay only requires that your SKU be unique across your active eBay listings (and not your ended listings).

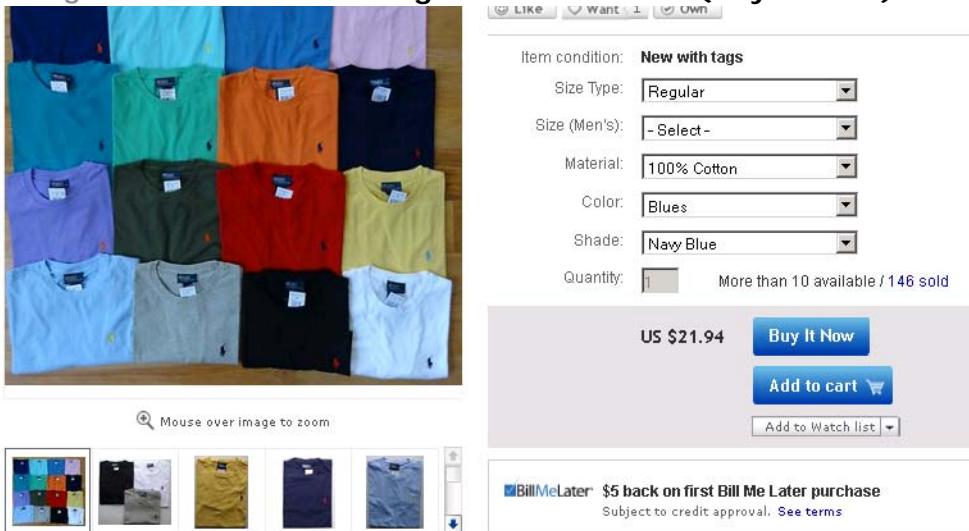
Configuring Variation-Selection Widgets for Buyers (View Item)

In addition to defining each variation, you need to define the set of all possible names and values that can be applicable to the listing (at any time in its life cycle). Provide this information in the [VariationSpecificsSet](#).

This information configures variation selection widgets that appear on eBay's View Item page. For example, if you specify Size (with options like "S" and "M") and Color (with options like "Blue" and "Pink") as names in the set, eBay's View Item page displays **Size** and **Color** drop-down lists to enable a buyer to choose a variation of interest (like medium pink).

[Figure 14-5](#) on page 274 shows a portion of the View Item page with drop-down lists that enable a buyer to choose a size and color combination.

Figure 14-5 View Item Page with Variations (Buyer View)



The order in which you specify the names and values also controls the order in which the selection widgets appear on the View Item page. For example, if you specify “Size”, then “Color”, and then “Shade” as names, the View Item page shows drop-down lists with those labels in that order. For “Size”, if you specify “XS”, “S”, “M”, “L”, “XL”, and “XXL” as values, the View Item page shows those values in that order in the Size drop-down list.

Note: eBay cannot reliably determine the order of the variation specifics based on the order in which you specify them in the variations themselves. This is because when you add a new variation (during a revise or relist activity) the new variation appears at the end of the list (regardless of where you specify it in your request). Over the lifetime of the listing, you may add new values, such as new colors or sizes, when you add variations. However, you can't modify the order in which names and values appear in existing variations.

Thus, you need to use [VariationSpecificsSet](#) to independently define the set of all possible names and values, and the order in which to present them to buyers.

eBay analyzes the listing's latest variation data (including [Quantity](#)) against this set, and formats the View Item widgets appropriately so that buyers can only select combinations that are currently available for sale. Inapplicable combinations may be displayed as read-only or hidden (depending on eBay's ongoing analysis of factors that lead to successful sales).

Each name must have at least one (1) value. You can specify up to 30 values for each name in the set. For example, for a name like Size, you could specify XS, S, M, L, XL, XXL, and XXXL (which would be 7 values).

The information in this set is also used to crosscheck the details that you specify in the listing's variations over the lifetime of the listing.

Use [GetCategorySpecifics](#) to retrieve recommendations for names, values, and order.

At a minimum, the set must include every name and value that appears in the listing's variations. In addition, it can include other values that are not yet available. For example, suppose your listing includes shirts in various sizes and colors, but all your yellow shirts are back-ordered. You can't specify yellow variations with a zero quantity when you first list the item. In this situation, you can still specify “Yellow” as one of the colors in the full set, and then revise the listing later to add the yellow variations when the items become available in your inventory.

If you add a new value in your variations, you also need to add it in [VariationSpecificsSet](#). To do this in [ReviseFixedPriceItem](#), pass in the entire [VariationSpecificsSet](#) node with the complete set of names and values for the listing.

Defining Pictures for Variations

A listing can have a shared (or default) set of pictures at the item level, and also pictures that help buyers visually distinguish the variations.

For the variations, you choose one variation specific name to classify (or associate with) all the variation pictures. For example, if the variations are differentiated by Color and Size, you would likely choose Color as the classifier for all the pictures. Call [GetCategorySpecifics](#) to determine which name most people use for the variation pictures in each category.

You can specify one set of pictures for each value that matches the selected name. For example, if the name is Color, and its possible values are Blue, Black, and Pink, you can specify a set of pictures for all the blue variations, another set for all the black variations, and another set for all the pink variations.

[Figure 14-6](#) on page 275 shows one way an application can enable seller to specify a shared set of pictures and additional sets of pictures for the variations (for each color, in this case).

Figure 14-6 Multi-Variation Listing with Pictures (Seller View)

The screenshot shows the eBay Seller View interface for managing a multi-variation listing. On the left, there are four main sections:

- Shared:** Shows a grid of 8 shared pictures. Buttons for "Add pictures" and "Remove all pictures" are at the top. Below the grid are navigation arrows and a "Your pictures: 9 | 3 can be added" message.
- Color: Pink:** Shows 2 pictures. Buttons for "Add pictures" and "Remove all pictures" are at the top. Below the grid are navigation arrows and a "Your pictures: 2 | 10 can be added" message.
- Color: Black:** Shows 1 picture. Buttons for "Add pictures" and "Remove all pictures" are at the top. Below the grid are navigation arrows and a "Your pictures: 1 | 11 can be added" message.
- Color: Blue:** Shows 1 picture. Buttons for "Add pictures" and "Remove all pictures" are at the top. Below the grid are navigation arrows and a "Your pictures: 1 | 11 can be added" message.

On the right, there is a large image of a red t-shirt labeled "What buyers see". Below it is a grid of smaller thumbnail images representing the different variations and their sizes.

All pictures in the item-level picture set and the variation picture sets must use the same type of picture hosting (either eBay Picture Services or self-hosted pictures). You can't have a mixture of both in the same listing.

The rules and logic for pictures (including Gallery pictures) at the item-level have not changed; they are the same as for any fixed-price item (for the same site and category).

Each picture set can contain the same maximum number of pictures that are allowed at the item level (for that site and category). For example, if you can specify up to 12 eBay Picture Services pictures in the shared set ([PictureDetails](#)), then you can specify up to 12 eBay Picture Services pictures in the pink picture set, 12 in the black picture set, and 12 in the blue picture set.

Picture fees are based on the number of pictures in the set with the most pictures. So, if the shared set has 2 pictures and the pink, black, and blue sets have 3 pictures each, then fees are based on 3 pictures. If you purchase the Picture Pack option for the shared pictures ([PictureDetails.PhotoDisplay=PicturePack](#)), the option is also used for the variation pictures for no additional fee.

The shared [GalleryType](#) (in [Item.PictureDetails](#)) also applies to all the variation-level gallery pictures. The first picture in each variation picture set is used as the gallery picture for that set. The actual gallery picture displayed to users depends on how they perform the search: If a user enters a general keyword search (e.g., for “pink Polo shirt”), the gallery picture is based on the item-level picture settings. If a user uses the left-hand navigation widgets on eBay’s search page and sets Color=Pink, the gallery picture is based on the first picture in the pink picture set.

Note: As the shared picture is shown in results for keyword searches, it can be helpful to use a picture that conveys information about multiple variations. For example, you could include several representative shirts in a single picture.

Pictures for variations are optional (but recommended for a better buyer experience). eBay displays all available pictures in the View Item page. The first shared (item-level) picture has initial focus in the View Item page. When a buyer chooses a variation, the focus changes to highlight the appropriate pictures. For example, if the buyer selects a blue shirt, the focus will change to the first blue picture. If you provide no pictures for the blue shirts, the shared picture will include messaging that there is no picture available for Blue.

You can revise the name associated with the pictures if the listing has no purchases and ends in more than 12 hours. If you delete the name, all the corresponding picture sets are also deleted.

If the listing has no purchases and ends in more than 12 hours, you can add more picture sets, and add or remove pictures from each set. You can also delete all the variation pictures.

If the listing has purchases or ends within 12 hours:

- You can add new picture sets
- You can add new eBay Picture Services pictures to existing picture sets
- You can’t modify picture sets that contain self-hosted pictures.

Revising and Relisting with Variations

This section describes rules for revising and relisting items that include variations.

- “[Changing the Category](#)” on page 277
- “[Adding Variations](#)” on page 277
- “[Deleting Individual Variations](#)” on page 277
- “[Deleting All Variations](#)” on page 278
- “[Modifying Existing Variations](#)” on page 278

Note that if any variation within a listing has at least one purchase, the listing’s *item*-level data is subject to the usual revision restrictions for listings with purchases.

Changing the Category

If you change the primary category of a listing with variations, the listing must adhere to all the rules (including item specifics rules) that apply to the new category. Otherwise, the listing request will fail.

When the primary category changes, you may need to specify the [Variations](#) node again. As eBay’s recommendations are different for each leaf category, we suggest that you give the seller the option to review and choose from recommended variation specifics for the new category.

When you change the category, the relisted item does not retain the popularity of the original listing.

Adding Variations

To add new variations when you revise or relist, pass in the new [Variation](#) nodes. (You do not need to pass in all the original variations.) See the [ReviseFixedPricelItem samples](#) for an example.

For example, if the listing already has variations for small-size blue, medium-size black, and large-size pink colors, you could add new variations for medium-size blue and large-size black colors. As a general rule, when you add a new variation, it appears at the *end* of the list of variations in [GetItem](#).

If you add new variation names or values in the new variations, be sure to also update existing variations as well as [VariationSpecificsSet](#). See “[Modifying Existing Variations](#)” on page 278.

Adding a variation when relisting does not affect the sales rank or popularity of the other variations in the listing.

Deleting Individual Variations

If you want to delete a variation that has purchases, set the [Quantity](#) to 0. You cannot fully delete a variation with purchases, because the order line item data needs to remain available to the buyer(s) and the seller.

If a variation has no purchases, delete it by set the [Variation.Delete](#) field to `true`. (Don't use [DeletedField](#).)

Note: [DeletedField](#) can only be used to delete a parent node or all instances of a child node (e.g., all instances of [Item.PictureDetails.PictureURL](#)). So, you can't use [DeletedField](#) to delete a single variation.

See the [ReviseFixedPriceltem samples](#) for an example.

The listing must retain at least one variation with a non-zero quantity to remain active.

Deleting a variation when relisting does not affect the sales rank or popularity of the other variations in the listing.

Deleting All Variations

To delete all variations from a listing (without replacing them with new variations at the same time):

- 1 End the listing by using [EndFixedPriceltem](#) or [EndItem](#).
- 2 Create a new listing without variations by using [AddFixedPriceltem](#) or [AddItem](#).

Modifying Existing Variations

When you modify variations, they are subject to the same rules as any new listing with variations (e.g., every variation in the relisted item must have the same set of names and a unique combination of values).

If [InventoryTrackingMethod](#) is set to SKU, you cannot modify a variation's SKU when you revise or relist.

This section covers these details:

- ["Adding Variation Specific Names" on page 278](#)
- ["Modifying Variation Specific Names" on page 278](#)
- ["Modifying Variation Values" on page 279](#)
- ["Modifying a Variation's Quantity when Revising" on page 279](#)
- ["Modifying a Variation's Quantity When Relisting" on page 280](#)

Adding Variation Specific Names

To add new variation specifics to active listings, pass in the entire set of variation specifics in a single request, and include the new detail for all of them. For example, you could add Strap Length to a handbag listing, with varying values like Shoulder, Wrist, etc.

Modifying Variation Specific Names

When you are revising a listing, you can't delete an existing variation specific name (such as Material) from any variations, but you can change the name (e.g., to Fabric) by using [ModifyNameList](#).

To change the name, specify the current name (e.g., Material) in [ModifyNameList.Name](#), and the new name (e.g., Fabric) in [ModifyNameList.NewName](#). If you are modifying the values, also be sure to

specify the new name and its modified values in [VariationSpecificsSet](#), along with the names you are not changing.

To ensure consistent results, also update [Variations.Pictures.VariationSpecificName](#) if any pictures are associated with the old name.

Note that you cannot change the name of a required item specific.

Modifying Variation Values

When you revise a listing, if you need to correct the spelling of a variation value, you first delete the variation, and then you add a new variation instead. For example, suppose you realize that a set of T-shirts that you listed with Color=Blue should use Color=Periwinkle. In this case, you would delete the variations with Color=Blue (or set their quantity to zero, if they're not eligible for deletion) and add new variations with Color=Periwinkle instead.

When you relist, you are creating a new listing, so you can change the text of the names and values. Use the same process that you use when revising listings. The relisted item is subject to the same rules as any new listing with variations (e.g., every variation in the relisted item must have the same set of names and a unique combination of values).

Modifying a Variation's Price

You can revise the price of a variation as long as the quantity is greater than zero (or if you increase the quantity to a non-zero value at the same time), even after it has purchases. This enables you to adjust the price competitively and to offer discounts or sales as needed.

To revise a variation's price, specify the variation to be changed in [ReviseFixedPriceltem](#), and include the revised price in [Variation.StartPrice](#) along with the other details of the variation. You only need to specify the variation you are changing. See the [ReviseFixedPriceltem samples](#) for an example.

Alternatively, if you use SKUs to track your listings, use [ReviselInventoryStatus](#) to quickly change the prices of variations across multiple listings. Specify the variation's SKU in [InventoryStatus.SKU](#), and the variation's price in [InventoryStatus.StartPrice](#).

Modifying a Variation's Quantity when Revising

You can revise the quantity of a variation, even after it has purchases (order line items).

To revise a variation's price, specify the variation to be changed in [ReviseFixedPriceltem](#), and include the revised quantity in [Variation.Quantity](#) along with the other details of the variation. You only need to specify the variation you are changing.

Alternatively, if you use SKUs to track your listings, use [ReviselInventoryStatus](#) to quickly change the quantities for variations across multiple listings. Specify the variation's SKU in [InventoryStatus.SKU](#), and the variation's quantity in [InventoryStatus.Quantity](#).

If you change a variation's quantity but items from it have already sold, eBay adds the quantity sold to the new quantity you specify. In other words, suppose you list with [AddFixedPriceltem](#) and set a variation's quantity to 10. A few days later, the quantity sold is 6. Now you increase your inventory and you use [ReviseFixedPriceltem](#) to set the variation's available quantity to 20. In [GetItem](#), the variation's quantity is returned as 26 ([Quantity + QuantitySold](#)). (This means the quantity available can still be calculated as [Quantity - QuantitySold](#).)

You can revise the quantity of any variation to zero (0), as long as at least one variation in the listing has a non-zero value. If you revise a variation's quantity to zero and the variation has never had purchases, eBay deletes the variation. If quantities of the variation have been sold, it can't be deleted (but the quantity can be zero). Also see “[Deleting Individual Variations](#)” on page 277 for other ways to delete a variation.

If the variation sells out, and then you restock your inventory, you can revise the same variation to add more quantity, as long as the listing is still active. For example, suppose you list 50 medium blue Polo shirts in one variation and 50 medium red shirts in another. You sell all 50 medium blue shirts (so the quantity available is now zero), but the red shirts are still available. A week later, you get 20 more blue shirts in stock. You can revise the medium blue shirt variation and specify a new [Variation.Quantity](#) of 20 as long as the listing is still active.

Note: For Best Match information related to multi-variation listings, see the Best Match information at the following topic: [Multi-quantity Fixed Price listings with variations](#).

Modifying a Variation's Quantity When Relisting

When you use [RelistFixedPriceltem](#), each variation's quantity retains its original number (not the quantity available) unless you specifically change it.

For example, suppose your listing includes a variation with 20 blue shirts. You sell 15, leaving 5 available. Then you end and relist the listing. By default, eBay assumes the quantity of the blue shirt variation should be set to 20 (the original quantity). Therefore, it's a good idea to set the correct quantity for each variation when you relist. Use the same process that you use for revising a variation's quantity (see “[Modifying a Variation's Quantity when Revising](#)” on page 279).

Retrieving a Listing's Variations

Use [GetItem](#) to retrieve variations (if any) that the seller included in the listing. (If the listing has no variations, the [Variations](#) node is not returned.)

eBay returns variations in the order that the seller specified them. We suggest you display them in that order, in case the seller grouped or ordered them in a logical way.

In addition, [GetItem](#) returns the full set of all possible variation specifics in [VariationSpecificsSet](#). If you have a buying application, you can use this information to configure drop-down lists or other widgets to help buyers choose the variation they prefer. See “[Configuring Variation-Selection Widgets for Buyers \(View Item\)](#)” on page 273 for more information.

Optionally, you can retrieve a subset of the available variations by specifying one or more variation specifics in the [GetItem](#) request. See the [GetItem Call Reference](#) for more information.

Buying Items in Multi-Variation Listings

In order to purchase items from a multi-variation listing, you need to specify which variation you are purchasing.

If you are using [PlaceOffer](#), you need to specify the [ItemID](#) and [VariationSpecifics](#) for the variation being purchased (otherwise, eBay can't uniquely identify the variation).

You can purchase multiple items (quantity) from the same variation at the same time, just as you would for any multi-quantity fixed-price item. However, you can't purchase items from two different variations at the same time (with the same purchase action). You can combine two order line items with variations into a Combined Payment order, as you can with any other order line items.

Retrieving Order Line Items with Variations

A single order line item can include multiple items from the same variation in a listing. A single order line item cannot include items from multiple variations, even if they are in the same fixed-price listing.

To retrieve order line items for variations, you can pass in one or more [SKU](#) nodes in the [SKUArray](#) container of the [GetSellerTransactions](#) request, or you can pass in one or more SKUs in the [ItemTransactionIDArray.ItemTransactionID](#) container(s) of the [GetOrderTransactions](#) request. Each [TransactionArray.Transaction.Variation](#) container in either one of these call responses will include information about the variation that was purchased, the quantity purchased, and other details about the variation. If you're a seller, this information helps you determine which variation to ship.

If your order management application does not handle variations, you can still use [TransactionPrice](#) and other order line item-level information. However, you (or the user) may need to contact the buyer or seller to understand which variation was purchased (and which one to ship).

Monitoring Listings and Orders with Variations

[GetMyeBayBuying](#), [GetMyeBaySelling](#), and related calls return variations for applicable listings. Only the variations that have activity are returned.

In a future release, [GetSellerEvents](#) will provide details about the variation tied to each applicable event.

Platform notifications include details about variations, when applicable.

Identifying Variations in Member Messages

Member-to-member communications (such as Ask Seller a Question) occur at the Item level, not at the variation level. The member messaging API calls don't provide structured variation details in the request or response.

Buyers may optionally choose to provide information about a variation of interest in their message text.

Chapter 15

Listing Items with Parts Compatibility

This chapter introduces *parts compatibility* and describes how it works with eBay listings. It provides details for listing items with parts compatibility information, how parts compatibility information affects item searches, and how to retrieve parts compatibility information.

Parts compatibility is an eBay feature that uses structured data to associate compatible assemblies with parts listed on eBay. For example, parts compatibility enables sellers to specify accurately and comprehensively the vehicles on which a side mirror or a rim will fit. Parts compatibility improves search relevancy and frees up item titles and descriptions for more useful descriptions of the part.

["Parts Compatibility Supported Categories" on page 282](#)

["Parts Compatibility Listing Flow" on page 285](#)

["Parts Compatibility Search Flow" on page 302](#)

["APIs for Parts Compatibility" on page 306](#)

Parts compatibility is supported in select Parts & Accessories categories for the eBay Motors site (site ID 100) and eBay Germany site (site ID 77) only.

Parts Compatibility Supported Categories

Parts compatibility is enabled by category. Use [GetCategoryFeatures](#) to determine which categories support parts compatibility, the type of compatibility supported ([BySpecification](#) or [ByApplication](#)), and/or the minimum required and maximum allowed compatible applications to provide when specifying parts compatibility by application.

["Checking Individual Leaf Categories" on page 283](#)

["Identifying All Categories that Support Parts Compatibility" on page 284](#)

Checking Individual Leaf Categories

If you want to check if a specific leaf category supports parts compatibility, submit a [GetCategoryFeatures](#) request with the category ID and the compatibility feature IDs as shown in the following sample:

Example 15-1 Check parts compatibility support for a leaf category, request

```
<?xml version="1.0" encoding="utf-8"?>
<GetCategoryFeaturesRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <RequesterCredentials>
        <eBayAuthToken>ABC...123</eBayAuthToken>
    </RequesterCredentials>
    <Version>661</Version>
    <CategoryID>33709</CategoryID>
    <FeatureID>CompatibilityEnabled</FeatureID>
    <FeatureID>MinCompatibleApplications</FeatureID>
    <FeatureID>MaxCompatibleApplications</FeatureID>
    <DetailLevel>ReturnAll</DetailLevel>
</GetCategoryFeaturesRequest>
```

In this case, the response indicates the category supports parts compatibility by application and that 41 is the maximum number of compatibilities that can be specified. Since [MinItemCompatibility](#) is not returned for the category, the default site setting, 0, applies. That is, you are not required to specify parts compatibility information when you list items in the category.

Example 15-2 Check parts compatibility support for a leaf category, response

```
<GetCategoryFeaturesResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2010-03-31T21:10:31.059Z</Timestamp>
    <Ack>Success</Ack>
    <Version>663</Version>
    <Build>E663_CORE_BUNDLED_10949798_R1</Build>
    <CategoryVersion>1037</CategoryVersion>
    <UpdateTime>2010-03-31T18:50:03.000Z</UpdateTime>
    <Category>
        <CategoryID>33709</CategoryID>
        <ItemCompatibilityEnabled>ByApplication</ItemCompatibilityEnabled>
        <MaxItemCompatibility>41</MaxItemCompatibility>
    </Category>
    <SiteDefaults>
        <ItemCompatibilityEnabled>Disabled</ItemCompatibilityEnabled>
        <MinItemCompatibility>0</MinItemCompatibility>
        <MaxItemCompatibility>0</MaxItemCompatibility>
    </SiteDefaults>
    <FeatureDefinitions>
        <ItemCompatibilityEnabled/>
        <MaxItemCompatibility/>
        <MinItemCompatibility/>
    </FeatureDefinitions>
</GetCategoryFeaturesResponse>
```

Identifying All Categories that Support Parts Compatibility

To identify categories in which you can list items with parts compatibility, submit a [GetCategoryFeatures](#) request with the parts compatibility feature IDs and [ViewAllNodes](#) set to true as shown in the following sample:

Example 15-3 Check parts compatibility support for a leaf category, request

```
<?xml version="1.0" encoding="utf-8"?>
<GetCategoryFeaturesRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <RequesterCredentials>
        <eBayAuthToken>ABC...123</eBayAuthToken>
    </RequesterCredentials>
    <Version>661</Version>
    <FeatureID>CompatibilityEnabled</FeatureID>
    <FeatureID>MinCompatibleApplications</FeatureID>
    <FeatureID>MaxCompatibleApplications</FeatureID>
    <ViewAllNodes>true</ViewAllNodes>
    <DetailLevel>ReturnAll</DetailLevel>
</GetCategoryFeaturesRequest>
```

This returns all categories that override the site defaults for parts compatibility. The [SiteDefaults](#) node in the response indicates that parts compatibility is disabled by default, and that the max and minimum compatibilities required for a listing are both 0 by default.

The first category in the response, 33707 (Lighting & Lamps), is a parent category. If any of its children (leaf) categories had overridden its settings for parts compatibility, they would have been returned with their settings. Since only the parent category is returned, all leaf nodes have inherited the parent's settings:

- Each leaf category under Lighting & Lamps, such as 33709 (Fog/Driving Lights) supports parts compatibility by application
- Maximum number of compatibilities that can be specified for an item listed in the leaf categories is 41
- Minimum number of compatibilities required is 0 (i.e., listing with parts compatibility is optional in the leaf categories)

Example 15-4 Check parts compatibility support for a leaf category, response

```
<GetCategoryFeaturesResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2010-03-31T21:00:01.412Z</Timestamp>
    <Ack>Success</Ack>
    <Version>663</Version>
    <Build>E663_CORE_BUNDLED_10949798_R1</Build>
    <CategoryVersion>1037</CategoryVersion>
    <UpdateTime>2010-03-31T17:53:38.000Z</UpdateTime>
    <Category>
        <CategoryID>33707</CategoryID>
        <ItemCompatibilityEnabled>ByApplication</ItemCompatibilityEnabled>
        <MinItemCompatibility>0</MinItemCompatibility>
        <MaxItemCompatibility>41</MaxItemCompatibility>
    </Category>
```

Example 15-4 Check parts compatibility support for a leaf category, response

```
<Category>
  <CategoryID>170578</CategoryID>
  <ItemCompatibilityEnabled>BySpecification</ItemCompatibilityEnabled>
</Category>
<Category>
  <CategoryID>170577</CategoryID>
  <ItemCompatibilityEnabled>BySpecification</ItemCompatibilityEnabled>
</Category>
<Category>
  <CategoryID>170583</CategoryID>
  <ItemCompatibilityEnabled>ByApplication</ItemCompatibilityEnabled>
  <MinItemCompatibility>1</MinItemCompatibility>
  <MaxItemCompatibility>300</MaxItemCompatibility>
</Category>
<SiteDefaults>
  <ItemCompatibilityEnabled>Disabled</ItemCompatibilityEnabled>
  <MinItemCompatibility>0</MinItemCompatibility>
  <MaxItemCompatibility>0</MaxItemCompatibility>
</SiteDefaults>
<FeatureDefinitions>
  <ItemCompatibilityEnabled/>
  <MaxItemCompatibility/>
  <MinItemCompatibility/>
</FeatureDefinitions>
</GetCategoryFeaturesResponse>
```

The remaining categories, 170577, 170578, and 170583, are test categories. Categories 170577 (Rims Test Category) and 170578 (Tires Test Category) both support parts compatibility by specification. In these categories, item specifics are used to specify parts compatibility.

Parts Compatibility Listing Flow

The [AddItem](#) family of calls now allows you to list with parts compatibility in three ways:

- List with an eBay catalog product that supports Parts Compatibility ([Item.ProductListingDetails.ProductReferenceID](#)). See “[Specifying Parts Compatibility by Product](#)” on page 286.
- Specify parts compatibility manually by entering the assemblies (vehicles for the currently supported categories), to which the part applies, as name-value pairs ([Item.ItemCompatibilityList](#)). See “[Manual Parts Compatibility by Application](#)” on page 290.
- Specify parts compatibility manually by specifying the part's relevant dimensions and characteristics (e.g., Section Width, Aspect Ratio, Rim Diameter, Load Index, and Speed Rating)

for a tire) using item specifics (`Item.ItemSpecifics`). See “[Manual Parts Compatibility by Specification](#)” on page 294.

Note: The Product API and Product Metadata API are required to determine which products support parts compatibility. Similarly, additional APIs are required to look up the property names and values for specifying what assembly (vehicle) a part applies.

- Specify parts compatibility manually, using a K type vehicle specification number (e.g., 25456). Listing with a K type vehicle number is supported on the eBay Germany (site ID 77), United Kingdom (site ID 3), and Australia (site ID 15) sites.

Sellers can use `GetItem` to confirm the number of compatibilities specified for a given listing (`Item.ItemCompatibilityCount`) or to retrieve all parts compatibility details for the item (`Item.ItemCompatibilityList`). See “[Specifying Parts Compatibility Manually](#)” on page 289.

“[Specifying Parts Compatibility by Product](#)” on page 286
“[Specifying Parts Compatibility Manually](#)” on page 289
“[Revising Parts Compatibilities for an Item](#)” on page 297
“[Retrieving Parts Compatibilities for Listed Items](#)” on page 299

Specifying Parts Compatibility by Product

The easiest way to list an item with parts compatibility is to list with an eBay catalog product that has compatibilities.

To list an item with parts compatibility, using a product:

- Call `findProducts` in the Product API to locate your product and its eBay Product ID (`ePID`).

The `ePID` value is used as input to `AddItem` to list your item by product. The `findProducts` call lets you search by keywords (`keywords`) and/or you can specify product filters (`propertyFilter`) to refine your results.

The following request and response samples show how to find a product and its `ePID`.

Example 15-5 `findProducts` request with `keywords` and `propertyFilter`

```
<?xml version="1.0" encoding="UTF-8"?>
<findProductsRequest xmlns="http://www.ebay.com/marketplace/marketplacecatalog/v1/
services">
  <productSearch>
    <invocationId>1234567890</invocationId>
    <categoryId>33707</categoryId>
    <keywords>headlight</keywords>
    <propertyFilter>
      <propertyName>Supplier Name</propertyName>
      <value>
        <text>
          <value>Maxzone Auto Parts</value>
        </text>
      </value>
    </propertyFilter>
  </productSearch>
</findProductsRequest>
```

Example 15-6 findProducts response, showing ePID

```

<findProductsResponse xmlns="http://www.ebay.com/Marketplace/MarketplaceCatalog/v1/
  services">
  <ack>Success</ack>
  <version>1.3.0</version>
  <timestamp>2010-04-28T02:29:30.796Z</timestamp>
  <productSearchResult>
    <products>
      <productIdentifier>
        <ePID>81391542</ePID>
        <productId>110562:2:2106:3215887382:238213712:3796824651c739bf59ecdd19a0f833b6:1:1
        :1:1400646370</productId>
      </productIdentifier>
      <productDetails>
        <propertyName>Type</propertyName>
        <value>
          <text>
            <value>Sealed Beams, Lamps, Flashers, Fuses</value>
          </text>
        </value>
      </productDetails>
      ...
    </products>
    <paginationOutput>
      <pageNumber>1</pageNumber>
      <entriesPerPage>20</entriesPerPage>
      <totalPages>78</totalPages>
      <totalEntries>1547</totalEntries>
    </paginationOutput>
    <invocationId>1234567890</invocationId>
  </productSearchResult>
</findProductsResponse>

```

- 2** Call **getProductCompatibilities** in the Product API to confirm the parts compatibilities for the product.

The following **getProductCompatibilities** request and response indicate the eBay catalog product with **ePID** value 74311312 does have compatibilities.

Example 15-7 getProductCompatibilities request, Ford make vehicles only

```

<getProductCompatibilitiesRequest xmlns="http://www.ebay.com/Marketplace/
  MarketplaceCatalog/v1/services">
  <productIdentifier>
    <ePID>74311312</ePID>
  </productIdentifier>
  <applicationPropertyFilter>
    <propertyName>Make</propertyName>
    <value>
      <text>
        <value>Ford</value>
      </text>
    </value>
  </applicationPropertyFilter>

```

Example 15-7 getProductCompatibilities request, Ford make vehicles only (Continued)

```
<sortOrder>
  <sortOrder>
    <propertyName>Model</propertyName>
    <order>Ascending</order>
  </sortOrder>
  <sortPriority>Sort1</sortPriority>
</sortOrder>
</getProductCompatibilitiesRequest>
```

Example 15-8 getProductCompatibilities response

```
<getProductCompatibilitiesResponse xmlns="http://www.ebay.com/marketplace/
marketplacecatalog/v1/services">
  <ack>Success</ack>
  <version>1.3.0</version>
  <timestamp>2010-04-28T02:48:08.129Z</timestamp>
  <paginationOutput>
    <pageNumber>1</pageNumber>
    <entriesPerPage>20</entriesPerPage>
    <totalPages>3</totalPages>
    <totalEntries>48</totalEntries>
  </paginationOutput>
  <compatibilityDetails>
    <productDetails>
      <propertyName>Engine</propertyName>
      <value>
        <text>
          <value>5.4L 330Cu. In. V8 GAS SOHC Naturally Aspirated</value>
        </text>
      </value>
    </productDetails>
    ...
    <notes>
      <noteDetails>
        <propertyName>Relation Per Car Quantity</propertyName>
        <value>
          <text>
            <value>1</value>
          </text>
        </value>
      </noteDetails>
    </notes>
  </compatibilityDetails>
  ...
</getProductCompatibilitiesResponse>
```

- 3 Use **AddItem** (or related listing call) to list your item using the product.

The **ePID** value from the **findProducts** call is plugged into the **ProductReferenceID** field (**ProductListingDetails.ProductReferenceID**) in the **AddItem** request.

Example 15-9 AddItem request, listing by product

```
<?xml version="1.0" encoding="utf-8"?>
<AddItemRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <Item>
    ...
    <PrimaryCategory>
      <CategoryID>33708</CategoryID>
    </PrimaryCategory>
    <ProductListingDetails>
      <ProductReferenceID>81391542</ProductReferenceID>
      <IncludePrefilledItemInformation>true</IncludePrefilledItemInformation>
    </ProductListingDetails>
    ...
  </Item>
</AddItemRequest>
```

Example 15-10 AddItem response, listing by product

```
<AddItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2010-04-28T05:19:48.152Z</Timestamp>
  <Ack>Success</Ack>
  <Version>665</Version>
  <Build>E665_CORE_BUNDLED_11079705_R1</Build>
  <ItemID>110045056625</ItemID>
  <StartTime>2010-04-28T05:19:47.355Z</StartTime>
  <EndTime>2010-05-05T05:19:47.355Z</EndTime>
  <Fees>
    ...
    <Fee>
      <Name>ListingFee</Name>
      <Fee currencyID="USD">0.5</Fee>
    </Fee>
    ...
  </Fees>
</AddItemResponse>
```

See “[Retrieving Parts Compatibilities for Listed Items](#)” on page 299 for information about confirming your parts compatibility.

Specifying Parts Compatibility Manually

Manual parts compatibility is used when your part does not correspond to a catalog product in the compatibility-enabled category in which you want to list.

Relationships between parts and products can be specified manually in three ways:

- By Specification—the relationship between the part and the product is represented by specifications for the part, such as dimensions.
- By Application—the relationship between the part and the product is specified explicitly.
- By K type vehicle number—a K type number represents one or more vehicles that will be used as compatibilities.

Manual Parts Compatibility by Application

To manually specify parts compatibility by application:

- 1 Call [GetCategoryFeatures](#) to determine if the category supports compatibility by application.

See “[Parts Compatibility Supported Categories](#)” on page 282 for more information on identifying categories that support parts compatibility by application.

- 2 Call [getCompatibilitySearchNames](#) in the Product Metadata API to get the property names needed to list compatibilities.

Set the `dataset` field value to “Searchable” to retrieve property names that can be used to search for corresponding values. Note the `propertyName` values in the response. The `displaySequence` values indicate the hierarchy of the data. For example, Engine is subordinate to Trim, which is subordinate to Model, and so on.

Example 15-11 `getCompatibilitySearchNames` request for category 33707

```
<?xml version="1.0" encoding="UTF-8"?>
<getCompatibilitySearchNamesRequest xmlns="http://www.ebay.com/marketplace/
marketplacecatalog/v1/services">
  <categoryId>33707</categoryId>
  <dataset>Searchable</dataset>
</getCompatibilitySearchNamesRequest>
```

Example 15-12 `getCompatibilitySearchNames` response for category 33707

```
<getCompatibilitySearchNamesResponse xsi:type="GetProductSearchNamesResponse"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.ebay.com/
  marketplace/marketplacecatalog/v1/services">
  <ack>Success</ack>
  <version>1.1.0</version>
  <timestamp>2010-04-28T18:24:51.737Z</timestamp>
  <categoryId>33707</categoryId>
  <properties>
    <propertyName>
      <propertyName>Year</propertyName>
      <propertyNameMetadata>
        <displaySequence>1</displaySequence>
      </propertyNameMetadata>
      <propertyDisplayName>Year</propertyDisplayName>
    </propertyName>
  </properties>
</getCompatibilitySearchNamesResponse>
```

Example 15-12 getCompatibilitySearchNames response for category 33707

```

<propertyName>
  <propertyName>Make</propertyName>
  <propertyNameMetadata>
    <displaySequence>2</displaySequence>
  </propertyNameMetadata>
  <propertyDisplayName>Make</propertyDisplayName>
</propertyName>
<propertyName>
  <propertyName>Model</propertyName>
  <propertyNameMetadata>
    <displaySequence>3</displaySequence>
  </propertyNameMetadata>
  <propertyDisplayName>Model</propertyDisplayName>
</propertyName>
<propertyName>
  <propertyName>Trim</propertyName>
  <propertyNameMetadata>
    <displaySequence>4</displaySequence>
  </propertyNameMetadata>
  <propertyDisplayName>Trim</propertyDisplayName>
</propertyName>
<propertyName>
  <propertyName>Engine</propertyName>
  <propertyNameMetadata>
    <displaySequence>5</displaySequence>
  </propertyNameMetadata>
  <propertyDisplayName>Engine</propertyDisplayName>
</propertyName>
<dataset>Searchable</dataset>
</properties>
</getCompatibilitySearchNamesResponse>

```

- 3** Call [getCompatibilitySearchValues](#) with names from the preceding step to complete the name-value pairs that will make up a compatibility.

To list with parts compatibility by application, you must include valid Year, Make, and Model values for a given application. Optionally, you can include Trim and Engine values.

- 3a** Call [getCompatibilitySearchValues](#) to retrieve compatibility search values for Year.
- 3b** Make the call to retrieve compatibility search values for Make (see [Example 15-13](#)).
- 3c** Make the call to retrieve compatibility search values for Model. Use a property filter for Make to retrieve the models for a specific make (see [Example 15-15](#)).
- 3d** Optionally, make the call to retrieve compatibility search values for Trim and Engine.

Example 15-13 getCompatibilitySearchValues request for makes

```

<?xml version="1.0" encoding="UTF-8"?>
<getCompatibilitySearchValuesRequest xmlns="http://www.ebay.com/marketplace/
marketplacecatalog/v1/services">
  <categoryId>33707</categoryId>
  <listFormatOnly>true</listFormatOnly>
  <propertyName>Make</propertyName>
</getCompatibilitySearchValuesRequest>

```

Example 15-14 getCompatibilitySearchValues response for makes

```
<getCompatibilitySearchValuesResponse xmlns="http://www.ebay.com/marketplace/
marketplacecatalog/v1/services">
  <ack>Success</ack>
  <version>1.1.0</version>
  <timestamP>2010-04-28T18:49:44.639Z</timestamP>
  <metadataVersion>4</metadataVersion>
  <propertyValues>
    <propertyName>Make</propertyName>
    <value>
      <text>
        <value>AC</value>
      </text>
    </value>
  </propertyValues>
  <propertyValues>
    <propertyName>Make</propertyName>
    <value>
      <text>
        <value>AM General</value>
      </text>
    </value>
  </propertyValues>
  <propertyValues>
    <propertyName>Make</propertyName>
    <value>
      <text>
        <value>Abarth</value>
      </text>
    </value>
  </propertyValues>
  <propertyValues>
    <propertyName>Make</propertyName>
    <value>
      <text>
        <value>Acura</value>
      </text>
    </value>
  </propertyValues>
  ...
  <propertyValues>
    <propertyName>Make</propertyName>
    <value>
      <text>
        <value>Yugo</value>
      </text>
    </value>
  </propertyValues>
  <propertyValues>
    <propertyName>Make</propertyName>
    <value>
      <text>
        <value>Zundapp</value>
      </text>
    </value>
  </propertyValues>
  <treeFormat>false</treeFormat>
</getCompatibilitySearchValuesResponse>
```

Example 15-15 getCompatibilitySearchValues request for models with a property filter for make

```
<?xml version="1.0" encoding="UTF-8"?>
<getCompatibilitySearchValuesRequest xmlns="http://www.ebay.com/marketplace/
marketplacecatalog/v1/services">
    <categoryId>33707</categoryId>
    <propertyFilter>
        <propertyName>Make</propertyName>
        <value>
            <text>
                <value>Acura</value>
            </text>
        </value>
    </propertyFilter>
    <listFormatOnly>true</listFormatOnly>
    <propertyName>Model</propertyName>
</getCompatibilitySearchValuesRequest>
```

- 4 List your part with an **ItemCompatibilityList**, consisting of name-value pairs (e.g., Year and 2007, Make and Acura, and Model and TL) that describe the compatibilities.

The following **ItemCompatibilityList** can be used all calls in the **AddItem** family, including **Reviseltem** and the fixed-prices calls.

Note: If **ItemCompatibilityList** contains a **Compatibility** with an invalid combination of name-value pairs, the invalid compatibility will be reported in the **Errors** node of the response. As long as **ItemCompatibilityList** contains at least one valid **Compatibility**, the call will succeed and the listing will include all valid item compatibilities. If **ItemCompatibilityList** contains no valid **Compatibility**, the call will fail.

Example 15-16 ItemCompatibilityList

```
...
<Item>
    <ItemCompatibilityList>
        <Compatibility>
            <NameValueList>
                <Name>Year</Name>
                <Value>2006</Value>
            </NameValueList>
            <NameValueList>
                <Name>Make</Name>
                <Value>Acura</Value>
            </NameValueList>
            <NameValueList>
                <Name>Model</Name>
                <Value>TL</Value>
            </NameValueList>
            <CompatibilityNotes>Fits for all trims and engines.</CompatibilityNotes>
        </Compatibility>
    </ItemCompatibilityList>
</Item>
```

Example 15-16 ItemCompatibilityList (Continued)

```
<Compatibility>
<NameValueList>
  <Name>Year</Name>
  <Value>2007</Value>
</NameValueList>
<NameValueList>
  <Name>Make</Name>
  <Value>Acura</Value>
</NameValueList>
<NameValueList>
  <Name>Model</Name>
  <Value>TL</Value>
</NameValueList>
<CompatibilityNotes>Fits for all trims and engines.</CompatibilityNotes>
</Compatibility>
<Compatibility>
<NameValueList>
  <Name>Year</Name>
  <Value>2008</Value>
</NameValueList>
<NameValueList>
  <Name>Make</Name>
  <Value>Acura</Value>
</NameValueList>
<NameValueList>
  <Name>Model</Name>
  <Value>TL</Value>
</NameValueList>
<CompatibilityNotes>Fits for all trims and engines.</CompatibilityNotes>
</Compatibility>
<Compatibility>
<NameValueList>
  <Name>Year</Name>
  <Value>2009</Value>
</NameValueList>
<NameValueList>
  <Name>Make</Name>
  <Value>Acura</Value>
</NameValueList>
<NameValueList>
  <Name>Model</Name>
  <Value>TL</Value>
</NameValueList>
<CompatibilityNotes>Fits for all trims and engines.</CompatibilityNotes>
</Compatibility>
</ItemCompatibilityList>
</Item>
...

```

See “[Retrieving Parts Compatibilities for Listed Items](#)” on page 299 for information about confirming your parts compatibility.

Manual Parts Compatibility by Specification

When you list an item with parts compatibility by specification, you use item specifics ([Item.ItemSpecifics](#)) to specify the part's relevant dimensions and characteristics. These specifications are used to match the part with compatible applications. For example, when listing tires, the parts

compatibility is specified by describing the tire's dimensions (e.g., Section Width, Aspect Ratio, Rim Diameter, Load Index, and Speed Rating for a tire) using item specifics.

To manually specify parts compatibility by specification:

- 1 Call [GetCategoryFeatures](#) to determine if the category supports compatibility by specification.

See “[Parts Compatibility Supported Categories](#)” on page 282 for information on how to check if a category supports parts compatibility.

- 2 Call [GetCategorySpecifics](#) to look up the item specifics that apply to the part you are listing.

Parts compatibility by specification is supported in categories for products like rims and tires.

Rims require item specifics for the following dimensions or characteristics, at a minimum, to list with parts compatibility:

- Bolt Pattern
- Offset
- Rim Diameter
- Rim Width

Tires require item specifics for the following dimensions or characteristics, at a minimum, to list with parts compatibility:

- Aspect Ratio
- Load Index
- Rim Diameter
- Section Width
- Speed Index

- 3 List your part, using item specifics to specify the parts compatibility details.

See “[Retrieving Parts Compatibilities for Listed Items](#)” on page 299 for information about confirming your parts compatibilities. For information about listing with item specifics, see “[Working with Custom Item Specifics](#)” on page 481 and “[Listing with Recommended Item Specific Names and Values](#)” on page 490.

Manual Parts Compatibility by KType

Using a K type vehicle number to add item compatibilities provides the best accuracy. When a K type vehicle number is used, the corresponding item compatibility includes Make, Model, Platform,

Type, Engine, and Production Period. For example, the K type vehicle number 25456 specifies the following vehicle information:

- Make: Audi
- Model: A4
- Platform: 8k2
- Type: 1.8 TFSI
- Engine (KW/PS/ccm): 88/120/1798
- Production Period: 2008/01- (years 2008, 2009, 2010, and 2011)

Note: A K type vehicle number typically corresponds to a specific production period (range of years in which the given vehicle was produced). If the production period covers 3 years, then the Compatibility counts as 3 of the maximum allowed item compatibilities. [GetItem](#) returns **ItemCompatibilityCount**, which can help you monitor the number of compatibilities specified for an item.

It is important to implement a counter into your application to prevent sellers from specifying more than the maximum number of compatibilities for an item. An [AddItem](#) request will fail if the number of compatibilities specified exceeds the maximum allowed.

The ability to specify parts compatibility with a K type vehicle number is only supported on the eBay Germany (site ID 77) site.

To manually specify parts compatibility using a K type vehicle number:

- 1 Identify the K type vehicle number that corresponds to the products (vehicles) to which your part is compatible.

K type is a vehicle specification numbering system provided by TecDoc Informations Systems. For more information about TecDoc and the K type numbering system, see the [TecDoc Portal](#).

- 2 List your part with an **ItemCompatibilityList**, consisting of name-value pairs for each K type vehicle number that applies.

The following **ItemCompatibilityList** can be used by all calls in the [AddItem](#) family, including [ReviselItem](#) and the fixed-prices calls. The **Name** must be set to "KType" (the value is not case sensitive) and **Value** is set to the K type vehicle number.

If **ItemCompatibilityList** contains a **Compatibility** with an invalid combination of name-value pairs, the invalid compatibility will be reported in the **Errors** node of the response. As long as **ItemCompatibilityList** contains at least one valid **Compatibility**, the call will succeed and the listing will include all valid item compatibilities. If **ItemCompatibilityList** contains no valid **Compatibility**, the call will fail.

Example 15-17 Specifying parts compatibility using a K type number

```
<ItemCompatibilityList>
  <Compatibility>
    <NameValueList>
      <Name>KType</Name>
      <Value>27959</Value>
    </NameValueList>
    <CompatibilityNotes>Passt alle ordnungen.</CompatibilityNotes>
  </Compatibility>
</ItemCompatibilityList>
```

Revising Parts Compatibilities for an Item

If you have listed an item in a category that supports parts compatibility by application, you can add or remove item compatibilities.

Note: You cannot add or remove compatibilities for items that were listed with a product.

Adding More Item Compatibilities

To add additional compatibilities, simply call **ReviseItem** (or comparable revise call that supports parts compatibility) and specify any new compatibilities with **Item.ItemCompatibilityList**. Duplicate compatibilities (**Item.ItemCompatibilityList.Compatibility**) are ignored.

Example 15-18 ReviseItem request to add a compatibility

```
<?xml version="1.0" encoding="utf-8"?>
<ReviseItemRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <Item>
    <ItemID>110045128198</ItemID>
    <ItemCompatibilityList>
      <Compatibility>
        <NameValueList>
          <Name>Year</Name>
          <Value>2003</Value>
        </NameValueList>
        <NameValueList>
          <Name>Make</Name>
          <Value>Acura</Value>
        </NameValueList>
        <NameValueList>
          <Name>Model</Name>
          <Value>TL</Value>
        </NameValueList>
        <CompatibilityNotes>Fits for all trims and engines.</CompatibilityNotes>
      </Compatibility>
    </ItemCompatibilityList>
  </Item>
</ReviseItemRequest>
```

Removing Individual Item Compatibilities

To remove an item compatibility, call **Reviseltem** (or comparable revise call that supports parts compatibility) and specify the compatibility you want to remove. Include **Delete** with a value of **true** in the compatibility.

Example 15-19 Reviseltem request to remove a compatibility

```
<?xml version="1.0" encoding="utf-8"?>
<RevisedItemRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <Item>
    <ItemID>110045128198</ItemID>
    <ItemCompatibilityList>
      <Compatibility>
        <Delete>true</Delete>
        <NameValueList>
          <Name>Year</Name>
          <Value>2003</Value>
        </NameValueList>
        <NameValueList>
          <Name>Make</Name>
          <Value>Acura</Value>
        </NameValueList>
        <NameValueList>
          <Name>Model</Name>
          <Value>TL</Value>
        </NameValueList>
        <CompatibilityNotes>Fits for all trims and engines.</CompatibilityNotes>
      </Compatibility>
    </ItemCompatibilityList>
  </Item>
</RevisedItemRequest>
```

Removing or Replacing All Item Compatibilities

To remove all compatibilities, call **Reviseltem** (or comparable revise call that supports parts compatibility) and set **ReplaceAll** (**Item.ItemCompatibilityList.ReplaceAll**) to true. If no new item compatibilities are specified, all item compatibilities are removed from the item. If new compatibilities are included with the **ReplaceAll** field, the new compatibilities will replace any existing item compatibilities.

Example 15-20 Reviseltem request to remove all compatibilities

```
<?xml version="1.0" encoding="utf-8"?>
<RevisedItemRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <Item>
    <ItemID>110045128198</ItemID>
    <ItemCompatibilityList>
      <ReplaceAll>true</ReplaceAll>
    </ItemCompatibilityList>
  </Item>
</RevisedItemRequest>
```

See “[Retrieving Parts Compatibilities for Listed Items](#)” on page 299 for information about confirming your parts compatiblities.

Retrieving Parts Compatibilities for Listed Items

When an item is listed with parts compatibility information, the compatibilities are displayed on the eBay site in a Compatibility tab on the View Item page. This allows buyers to confirm compatibilities before purchasing an item.

Items listed with parts compatibility by application, either manually or with a product, automatically populate the table with applicable compatibilities, either with compatibilities that match the buyer’s search or a complete, paginated list of compatibilities. Items listed with parts compatibility by specification, let you search for compatibilities, using the Year, Make, and Model dropdowns.

Year	Make	Model	Trim	Engine	Notes
2008	BMW	M3	Base Convertible 2-Door	4.0L 3999CC V8 GAS DOHC Naturally Aspirated	View
2008	BMW	M3	Base Coupe 2-Door	4.0L 3999CC V8 GAS DOHC Naturally Aspirated	View
2008	BMW	M3	Base Sedan 4-Door	4.0L 3999CC V8 GAS DOHC Naturally Aspirated	View

Retrieving Compatibilities for Items Listed with Compatibilities by Application

To retrieve the compatibilities for items listed with parts compatibilities manually by application, call `GetItem` with the `IncludeItemCompatibilityList` field set to true.

Note: `GetItem` will not return compatibility information for items listed with parts capability using a K type vehicle number.

Example 15-21 GetItem request, including item compatibilities

```
<?xml version="1.0" encoding="utf-8"?>
<GetItemRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>AVC...123</eBayAuthToken>
  </RequesterCredentials>
  <ErrorLanguage>en_US</ErrorLanguage>
  <WarningLevel>High</WarningLevel>
  <Version>663</Version>
  <ItemID>110045056625</ItemID>
  <IncludeItemCompatibilityList>true</IncludeItemCompatibilityList>
  <DetailLevel>ReturnAll</DetailLevel>
</GetItemRequest>
```

Example 15-22 GetItem response, showing compatibilities

```
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2010-04-28T05:12:43.380Z</Timestamp>
  <Ack>Success</Ack>
  <Version>665</Version>
  <Build>E665_CORE_BUNDLED_11079705_R1</Build>
  <Item>
    ...
    <ItemID>110045056625</ItemID>
    ...
    <ItemCompatibilityList>
      <Compatibility>
        <NameValueList>
          <Name>Year</Name>
          <Value>2001</Value>
        </NameValueList>
        <NameValueList>
          <Name>Make</Name>
          <Value>Ford</Value>
        </NameValueList>
        <NameValueList>
          <Name>Model</Name>
          <Value>F-150</Value>
        </NameValueList>
        <NameValueList>
          <Name>Trim</Name>
          <Value>Harley-Davidson Edition Crew Cab Pickup 4-Door</Value>
        </NameValueList>
        <NameValueList>
          <Name>Engine</Name>
          <Value>5.4L 330Cu. In. V8 GAS SOHC Naturally Aspirated</Value>
        </NameValueList>
        <NameValueList/>
        <CompatibilityNotes>Relation Per Car Quantity:1<br></CompatibilityNotes>
      </Compatibility>
      ...
    </ItemCompatibilityList>
  </Item>
</GetItemResponse>
```

Finding Compatibilities by Specification

To find compatibilities for an item listed with parts compatibility by specification, call [findCompatibilitiesBySpecification](#) in the Product API. You must provide a [specification](#) for each of the dimensions or characteristics required to list the item. For example, if you are searching for compatibilities for a tire, you must provide [specification](#) values for Aspect Ratio, Load Index, Rim Diameter, Section Width, and Speed Index. Use [getProductSearchNames](#) to retrieve valid property names for use in the [specification](#) field.

To constrain the result set, specify a **compatibilityPropertyFilter**.

Example 15-23 **findCompatibilitiesBySpecification** request for tires

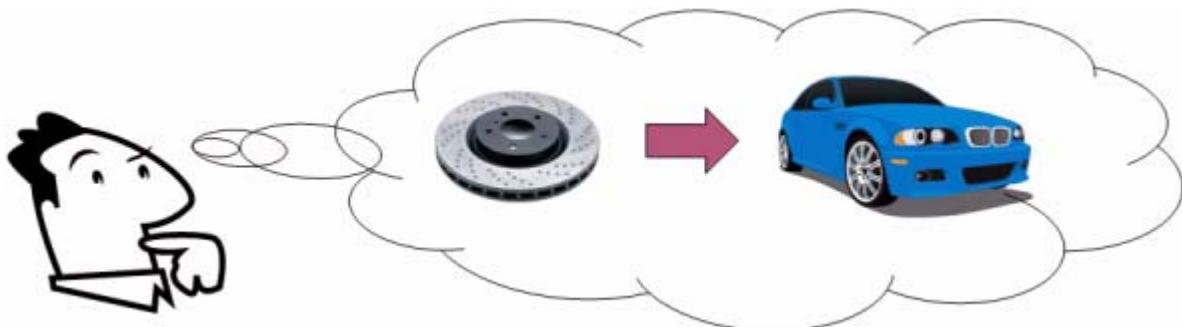
```
<?xml version="1.0" encoding="utf-8"?>
<findCompatibilitiesBySpecificationRequest xmlns="http://www.ebay.com/marketplace/
marketplacecatalog/v1/services">
    <categoryId>170578</categoryId>
    <compatibilityPropertyFilter>
        <propertyName>Make</propertyName>
        <value>
            <text>
                <value>Honda</value>
            </text>
        </value>
    </compatibilityPropertyFilter>
    <compatibilityPropertyFilter>
        <propertyName>Model</propertyName>
        <value>
            <text>
                <value>Accord</value>
            </text>
        </value>
    </compatibilityPropertyFilter>
    <compatibilityPropertyFilter>
        <propertyName>Year</propertyName>
        <value>
            <text>
                <value>2006</value>
            </text>
        </value>
    </compatibilityPropertyFilter>
    <specification>
        <propertyName>Section Width</propertyName>
        <value>
            <text>
                <value>225.0</value>
            </text>
        </value>
    </specification>
    <specification>
        <propertyName>Aspect Ratio</propertyName>
        <value>
            <text>
                <value>45.0</value>
            </text>
        </value>
    </specification>
    <specification>
        <propertyName>Rim Diameter</propertyName>
        <value>
            <text>
                <value>18.0</value>
            </text>
        </value>
    </specification>
```

Example 15-23 findCompatibilitiesBySpecification request for tires (Continued)

```
<specification>
  <propertyName>Load Index</propertyName>
  <value>
    <text>
      <value>91.0</value>
    </text>
  </value>
</specification>
<specification>
  <propertyName>Speed Rating</propertyName>
  <value>
    <text>
      <value>H</value>
    </text>
  </value>
</specification>
<datasetPropertyName>Make</datasetPropertyName>
<datasetPropertyName>Model</datasetPropertyName>
<datasetPropertyName>Year</datasetPropertyName>
<datasetPropertyName>Trim</datasetPropertyName>
<datasetPropertyName>Engine</datasetPropertyName>
</findCompatibilitiesBySpecificationRequest>
```

Parts Compatibility Search Flow

Parts buyers often think of parts in relation to the vehicle they fit (i.e., “I need a new brake disc for my BMW.”).



They search on ebay the same way:



Items listed with parts compatibility information help connect buyers with parts and accessories for their assembly, such as a car or truck. Parts compatibility lets sellers enter compatible vehicles in a structured place outside the title. This parts compatibility information is indexed for keyword searches, so buyers can find the part by its compatible vehicles, not just the part description.

["Searching By Application"](#) on page 303

["Searching By Specification"](#) on page 305

Searching By Application

When a buyer is looking for a part or accessory for his car, he can specify the year, make, and model of his car (sometimes referred to as compatibility search names) along with any details about the part he's trying to find (e.g., new headlight 2008 bmw m3). Listings with parts compatibility information get a best match boost in search results. Items in the search results with parts compatibility

information that matches terms in the buyer's search query display the compatibility match (e.g., Fits: 2008 BMW M3).

The screenshot shows the eBay Motors search interface. The search bar at the top contains the query "new headlight 2008 bmw m3". Below the search bar, there are options to "Find" or "Search" and a dropdown for "All of eBay Motors". A checkbox for "Include title and description" is checked. The search results page displays 165 matches found for the query. The results are filtered by "All items". There are three tabs: "List view [Customize]" (selected), "Picture view", and "Sort by". The first result is for "OSRAM SYLVANIA OEM HID XENON XENARC HEADLIGHT BULB D1S", which is a top-rated seller. The compatibility information "Fits: 2008 BMW M3" is highlighted with a red circle. The second result is for "HID Xenon Headlights D1C D1S D1R Super Deep Blue 3000K", also a top-rated seller, with the compatibility information "Fits: 2008 BMW M3" visible.

The Finding API supports compatibility in the same way as the site. The **keywords** field can include vehicle details, such as year, make, and model (e.g., 2008 bmw m3), in addition to terms for the part or accessory. When there are matches, and the item was listed with compatibility information, the **compatibility** field in the response returns terms from the query that match values in one of the item's

compatible applications (vehicles). If the query matches more than one compatible application for an item, the **compatibility** field returns “Two or more of your vehicles” to indicate multiple matches.

Example 15-24 Parts compatibility (by application) in search results

```
<findItemsByKeywordsResponse xmlns="http://www.ebay.com/marketplace/search/v1/services">
  <ack>Success</ack>
  <version>1.3.0</version>
  <timestamp>2010-04-27T23:54:06.800Z</timestamp>
  <searchResult count="100">
    <item>
      ...
      <compatibility>2008 BMW M3</compatibility>
      <condition>
        <conditionDisplayName>New</conditionDisplayName>
      </condition>
    </item>
    ...
  </searchResult>
  <paginationOutput>
    <pageNumber>1</pageNumber>
    <entriesPerPage>100</entriesPerPage>
    <totalPages>2</totalPages>
    <totalEntries>130</totalEntries>
  </paginationOutput>
</findItemsByKeywordsResponse>
```

Searching By Specification

Some items, such as tires or rims, are listed with parts compatibility by specification. If a buyer is searching for a new tire, he may read the specifications off his existing tire (e.g., section width, aspect ratio, and rim diameter) and use these specifications to search for a compatible replacement. For example, he may search for “michelin exalto 195 65 15,” where 195 is the section width, 65 is the aspect ratio, and 15 is the rim diameter. Items in the search results with parts compatibility information that matches terms in the buyer’s search query display the specification match (e.g., Specification: 195/65R15).

The screenshot shows an eBay search interface. The search term 'michelin exalto 195 65 15' is entered in the 'Find' field. The search results page indicates '1 match found for: michelin exalto 195 65 15'. The listed item is 'Michelin Pilot Exalto' with the specification '195/65R15' highlighted. Refinement options like 'Categories' (Parts & Accessories), 'Price', and 'Make' are visible on the left.

To search for items listed with compatibility information by specification, include the part specifications, such as a tire's section width, aspect ratio, and rim diameter (e.g., 225 45 15), in the **keywords** in addition to other terms for the part or accessory (e.g., <keywords>michelin exalto 195 65 15</keywords>). When there are matches, and the item was listed with compatibility information by specification, the **compatibility** field in the response returns terms from the query that match the items compatibility specifications.

Example 15-25 Parts compatibility (by specification) in search results

```
<findItemsByKeywordsResponse xmlns="http://www.ebay.com/marketplace/search/v1/services">
  <ack>Success</ack>
  <version>1.3.0</version>
  <timestamp>2010-04-28T00:33:52.106Z</timestamp>
  <searchResult count="1">
    <item>
      <itemId>110045053160</itemId>
      <title>Michelin Pilot Exalto</title>
      ...
      <compatibility>195/65R15</compatibility>
    </item>
    ...
  </searchResult>
  <paginationOutput>
    <pageNumber>1</pageNumber>
    <entriesPerPage>100</entriesPerPage>
    <totalPages>1</totalPages>
    <totalEntries>1</totalEntries>
  </paginationOutput>
</findItemsByKeywordsResponse>
```

This search behavior is supported by the Finding API.

APIs for Parts Compatibility

Parts compatibility is supported by the following APIs:

- Trading API
- Product Services: Product API and Product Metadata API
- Search API: Finding API

Trading API

Parts compatibility is supported in select Parts & Accessories categories for eBay Motors (site ID 100) only. The Trading API provides the calls you need to identify the categories that support parts compatibility, list (or revise) items with parts compatibility, and retrieve item compatibility information for items listed with parts compatibility.

Use [GetCategoryFeatures](#) to determine which categories on the eBay Motors US site support parts compatibility. See “[Parts Compatibility Supported Categories](#)” on page 282 for more information.

Listing items with parts compatibility is supported by the following:

- [AddFixedPricelItem](#)
- [AddItem](#)
- [AddItemFromSellingManagerTemplate](#)
- [AddItems](#)
- [AddSellingManagerTemplate](#)
- [RelistFixedPricelItem](#)
- [RelistItem](#)
- [ReviseFixedPricelItem](#)
- [ReviselItem](#)
- [ReviseSellingManagerTemplate](#)
- [VerifyAddlItem](#)
- [VerifyRelistItem](#)

Use [GetItem](#) to retrieve parts compatibility information for an active item listing.

See “[Parts Compatibility Listing Flow](#)” on page 285 for more information.

Trading API Call Reference

<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/index.html>

Product Services

eBay Product Services consists of two separate APIs, the Product API and Product Metadata API. Product Services provides the information sellers need to list items with parts compatibility by application. Items can be listed with parts compatibility by application either using a product that has compatibilities or by manually specifying the compatibilities.

See “[Parts Compatibility Listing Flow](#)” on page 285 for more information on how the Product API and the Product Metadata API support listing items with parts compatibility.

Product API

The Product API consists of calls that return product information. For example, you can use the Product API to identify products that have associated compatibilities. Listing an item with a product that has compatibilities (e.g., specific vehicle applications, such as a 2004 Honda Accord) is one way to list an item with parts compatibility information.

Product API Call Reference

<http://developer.ebay.com/DevZone/product/CallRef/index.html>

Product Metadata API

The Product Metadata API consists of calls that return product metadata for specific categories. For example, you can use the Product Metadata API to retrieve the compatibility search names (e.g., Make and Model) and values (e.g., Honda and Accord) that are allowed for a given category. You can use this compatibility metadata to construct compatible applications ([Item.ItemCompatibilityList](#)) for a specific category. Listing an item with compatible applications (e.g., a fender that fits specific cars) is one way to list an item with parts compatibility information. This is known as listing with compatibility by application.

Product Metadata API Call Reference

<http://developer.ebay.com/DevZone/product-metadata/CallRef/index.html>

Search API

The Finding API supports parts compatibility. When buyers search for parts with **keywords** queries that include compatibility information (e.g., vehicle year, make, or model) or part specifications (e.g., tire section width, aspect ratio, or rim diameter), the queries are run against the parts compatibility information. When there are matches, the matching compatibility or specification terms are returned with the matching item in a **compatibility** field.

Although parts compatibility is supported for listing items on the eBay Motors US site only. Parts compatibility for searching is supported on the both the core eBay US site and the eBay Motors US site.

See “[Parts Compatibility Search Flow](#)” on page 302 for more information on how parts compatibility is supported in item searches.

Finding API

<http://developer.ebay.com/DevZone/finding/CallRef/index.html>

Best Match API

<http://developer.ebay.com/DevZone/best-match-item-details/CallRef/index.html>

Chapter 16

Want It Now

On the eBay site, *Want It Now* is where buyers post messages describing something they would like to buy. A buyer may post to Want It Now because he is having difficulty finding a specific item or type of item. Buyers may also post to Want It Now because they want a specific quantity.

Sellers search Want It Now posts to find buyers for the items they have to offer. When a seller finds a post that closely describes an item she is selling on eBay, she responds to the Want It Now post with the item listing. The buyer then receives an email with a link to the listing. Sellers may also want to search Want It Now posts to gauge demand for certain items or to get ideas for items to sell on eBay.

The Want It Now calls give sellers the ability to programmatically search for and respond to Want It Now posts. There are three Want It Now API calls: [GetWantItNowSearchResults](#), [GetWantItNowPost](#), and [RespondToWantItNowPost](#). Each of these calls corresponds to a basic step for working with Want It Now posts: the seller runs a query to retrieve a list of posts that match keywords and/or categories corresponding to the types of items she sells, the seller gets individual posts to review their description before responding, and the seller responds to individual posts that are good matches for her items on eBay.

Retrieving Want It Now Posts

Use [GetWantItNowSearchResults](#) to retrieve Want It Now posts that are possible candidates for the seller. Enter a search string in the [Query](#) input field to specify a free-text keyword search. Enter a category number in the [CategoryID](#) input field to retrieve only posts from a specific category. These two fields are optional, they can be used in combination or separately. When both [Query](#) and [CategoryID](#) are specified, posts are returned only when they match both criteria. The optional [SearchWorldwide](#) and [SearchInDescription](#) input fields can be used to refine or expand searches.

[GetWantItNowSearchResults](#) returns a [WantItNowPostArray](#), which contains a [WantItNowPost](#) object for each post found matching the specified search criteria. Each [WantItNowPost](#) returned with [GetWantItNowSearchResults](#) contains the following data: [CategoryID](#), [PostID](#), [StartTime](#), [ResponseCount](#), [Site](#), and [Title](#). The [Description](#) for a post is not returned with this call.

Searching for Posts with a Query

You can specify a search string in the optional [Query](#) field of [GetWantItNowSearchResults](#) to retrieve Want It Now posts that match certain keywords. When a [Query](#) is specified, [GetWantItNowSearchResults](#) searches for Want It Now posts where the keywords are found in either the title or description, returning those posts in the result set. The value specified for [Query](#) can contain one or more keywords. You can specify AND or OR logic by including certain modifiers. Be careful when using spaces before or after modifiers.

EBay is deprecating wildcard (*) queries from all search tools and functions, including Web and mobile search, saved searches, and searches made through API calls. For more information, refer to the [Wildcards in eBay Searches deprecated product update page](#).

By default, both the title and description of Want It Now posts are searched, and posts are returned if the search string criteria is met for either. You can limit your query to just titles by adding `SearchInDescription` and setting it to `false`.

For more information about constructing a keyword query, including a table describing the operators and syntax used in keyword queries, see [Searching by Keywords](#).

Searching for Posts by Category

You can specify a category number in the optional `CategoryID` field of [GetWantItNowSearchResults](#) to retrieve Want It Now posts in the specified category and any leaf categories.

Searching All eBay Sites for Posts

By default, the search is limited to the site specified in the URL query string when the call is made. You can expand your query to all eBay sites by adding `SearchWorldwide` in the request and setting it to `true`.

Paginating Search Results

Use the [Pagination](#) property and its child elements to break the items matching the search criteria down into smaller subsets, or “pages” of data. `Pagination.EntriesPerPage` specifies the maximum number of posts to return in any given call. `Pagination.PageNumber` specifies which “page” of data to return in the current call. If no pagination values are specified, `Pagination.EntriesPerPage` defaults to 100 and `Pagination.PageNumber` defaults to 1.

There are corresponding fields in the [GetWantItNowSearchResults](#) response object that are used for paginating data:

- `PaginationResult.TotalNumberOfEntries` indicates the total number of posts matching the input search criteria that could be returned (with one or multiple calls). If the value specified in `Pagination.EntriesPerPage` is less than that returned in `PaginationResult.TotalNumberOfEntries`, then it will take more than one call to retrieve all of the matching posts.
- `PaginationResult.TotalNumberOfPages` indicates the total number of calls that would be required to retrieve all of the matching posts. The value specified in `Pagination.PageNumber` of the [GetWantItNowSearchResults](#) request indicates where in the sequence of multiple calls the application is, relative to the number returned in `PaginationResult.TotalNumberOfPages`.

Depending upon the size of the result set, developers may want to process the results of the call to exclude Want It Now posts that aren't applicable. For example, a developer might use `StartTime` values to eliminate posts that have been considered previously.

Retrieving All Data for a Single Post

Use `GetWantItNowPost` to get all the content for an individual Want It Now post, including `Description`. A seller should review the post description to ensure her item is a good match before responding to a post. `GetWantItNowPost` takes only the post number, `PostID`, as an input.

`GetWantItNowPost` returns a `WantItNowPost` object containing the following data: `CategoryID`, `Description`, `PostID`, `StartTime`, `ResponseCount`, `Site`, and `Title`. Sales representatives can review the descriptions for individual Want It Now posts to determine which item listings best match the post and whether or not to respond. Alternatively, a sales representative can view the Want It Now post on the site to retrieve additional information, such as pictures and what items have already been submitted in response to the post.

Responding to Want It Now Posts

Once the seller determines that a post is a good match for an item listed on eBay, she can use `RespondToWantItNowPost` to respond to the post with the item listing. `RespondToWantItNowPost` takes the post number, `PostID`, and the item number, `ItemID`, as an input. When a response is submitted, the buyer receives an email with a link to the specified item listing.

Responses appear on the Want It Now post page, with the item title, the price of the item, the number of bids on the item, and the time left before the listing ends. If the item has a picture, the picture is also included on the Want It Now post page. It may take several hours for a response to appear on the Want It Now post page after it has been submitted.

Attempting to respond to a post with the same item more than once results in an error. Sellers may want to track what items have submitted in responses to prevent errors.

Chapter 17

Shipping

This chapter discusses the following shipping topics:

- “Introduction to Shipping” on page 312
- “Working with Shipping in API Calls” on page 319
- “Specifying Locations to Where You Ship” on page 324
- “Specifying Shipping Services” on page 327
- “Specifying Shipping Types and Costs” on page 336
- “Specifying Other Costs” on page 350
- “Determining Shipping Costs for a Listing” on page 352
- “Other Shipping Features” on page 355

Introduction to Shipping

- “Domestic and International Shipping” on page 312
- “Shipping Carriers and Services” on page 314
- “Shipping Types” on page 315
- “Shipping Costs” on page 316
- “Participation in the Global Shipping Program” on page 317
- “Additional Resources” on page 319

Domestic and International Shipping

“Domestic” shipping is shipping within the same country as the listing site. “International” shipping is shipping to countries other than the country of the listing site. In the eBay API, there are domestic and international shipping services for each country that eBay supports. Call `GeteBayDetails` with the `DetailName` input filter set to `ShippingServiceDetails` to view valid shipping services for each country.

International Shipping with the Global Shipping Program

Sellers can handle international shipping themselves, or by using eBay’s *Global Shipping Program*. Global Shipping is a feature currently available to sellers who list items on the eBay US site (sellers can be registered anywhere). It enables a seller to sell an item to anyone, then use eBay to easily ship it from a US location to the buyer’s country (subject to restrictions on certain product categories, countries of manufacture and destination countries). The Global Shipping Program promotes international trade by providing express shipping with end-to-end tracking at significantly low shipping cost.

Note: Sections of this chapter that discuss international shipping features do not apply to the Global Shipping Program except where noted as such. However, sections that discuss domestic shipping features do apply to listings and orders that use the Global Shipping Program, except as noted.

eBay intends this program to achieve two primary goals:

- Sellers should be able to sell to buyers in other countries as easily as they can sell to domestic buyers, without any additional concern for the complexities of international shipping.
- Buyers should be able to choose from the largest possible selection of products available for purchase on eBay, regardless of either the buyers' or the sellers' countries.

Note: The current implementation of the Global Shipping Program is one stage in the process of accomplishing these goals. Over time, increasing seller participation in Global Shipping is expected to result in fewer logistical limitations and more shipping options.

Following is a comparison of conventional international shipping and the Global Shipping Program:

Table 17-1 International Shipping vs. Global Shipping Program

	International Shipping	Global Shipping Program
Benefits	<p>A seller can ship virtually any item to any destination, subject to governmental rules about shipping certain types of products and shipping to certain countries.</p> <p>The seller can choose from a variety of available international shipping services, price levels and associated features.</p>	<p>The seller is free to deal with just the domestic portion of the shipment without concern about the international portion.</p> <p>An <i>international shipping provider</i> (a company contracted by eBay) is completely responsible for the international portion of the shipment, which includes assessing costs, handling customs issues, providing insurance, and completing delivery to the buyer.</p> <p>The buyer pays all costs associated with the shipment at checkout, and will see no unexpected charges after that point.</p>
Limitations	<p>The seller is responsible for managing the logistics of the entire shipment, for insuring the shipment in a manner appropriate to the country and the carrier, and for ensuring delivery to the destination.</p>	<p>Some item categories, shipping methods and destination countries might not be supported by the international shipping provider for logistical, regulatory or cost reasons.</p>

All of the costs associated with the domestic leg of Global Shipping can be handled by the seller in the same way as regular domestic shipping. The costs associated with the international leg are determined by the international shipping provider, and are automatically included by eBay in the buyer's checkout process.

Note: The shipping and handling charges for the domestic and international legs are combined and presented to buyers as a single shipping charge in each listing, so the buyers have no direct exposure to the concept of an international shipping provider.

eBay encourages sellers to opt in to the Global Shipping Program. It will eventually become the default international shipping option, first for U.S. based sellers, and later for all sellers.

Sellers are presented with the option to join the Global Shipping Program on various pages of the eBay US site. Sellers will see this offer in promotional emails, at sign-in, on their My eBay page, and in other locations. Opting in to the program is done only from the web UI; there's no way to opt in using an API. Once opted in, sellers need not opt out again. They can simply choose not to offer the Global Shipping option in listings.

Shipping Carriers and Services

Generally, “shipping service” refers to a particular service offered by a shipping carrier. An example of a shipping carrier is the U.S. Postal Service. It offers a service called Priority Mail.

The eBay API has an enumerated type that defines many of the shipping services offered by shipping carriers with whom eBay has a relationship. For example, there is an enumeration value named **USPSPriority**, which corresponds to the previously mentioned Priority Mail service.

Note: For the domestic leg of a Global Shipping Program shipment, sellers can use any carrier and service that is available to them for U.S. domestic shipping.

Shipping Service Categories

The supported shipping services for the site are grouped into several categories. Shipping service categories include Economy, Standard, Expedited, One-Day Service, and Other. A shipping service is classified into a category based on how the shipping time range provided by the service aligns with the shipping time range defined for the category.

Note: In the eBay Seller Information Center, shipping service categories are referred to as shipping service “levels.” This User’s Guide will use the term “category” to align with the terminology used in the API.

You cannot specify a **ShippingCategory** in the Add/Revise/Relist family of API calls, as eBay does this categorization behind the scenes. **ShippingCategory** is read-only data that is returned in the **ShippingServiceDetails** and **ShippingCategoryDetails** container of **GeteBayDetails**. One possible use of **ShippingCategoryDetails** is to sort shipping services by **ShippingCategory** and then populate a pick list.

Shipping Services

Call **GeteBayDetails** with the **DetailName** input filter set to **ShippingServiceDetails** to view valid shipping services for each country.

Each instance of **ShippingServiceDetails.ShippingService** returned contains the name of a shipping service, but the indicated shipping service is currently valid for listing items only if the boolean field **ShippingServiceDetails.ValidForSellingFlow** is true.

Global Shipping Services

With the Global Shipping Program, the seller can use any domestic shipping category and service for the domestic leg of the shipment. The name of the selected service doesn’t appear to the buyer, and its cost is incorporated into the overall shipping cost for Global Shipping.

For the international leg of the shipment, Global Shipping has no equivalent to the domestic shipping categories; the level of service is determined by the international shipping provider, which specifies an international shipping service to complement the domestic shipping service chosen by the seller.

Note: Currently, eBay provides a single Global Shipping Program service, *International Priority Shipping*.

Subsequent to creating a listing with Global Shipping Program as the default international shipping option, Get calls will return **InternationalPriorityShipping** as the value of **ShippingDetails.InternationalShippingServiceOption.ShippingService**, and

`GlobalShipping_MultiCarrier` (the international shipping provider) as the international carrier name in the `ShippingDetails.ShipmentTrackingDetails.ShippingCarrierUsed` field.

Although no direct comparison can be made between domestic and international shipping speed, International Priority Shipping is intended to be viewed as a cost-effective international equivalent to the Expedited category of domestic services.

Specifying Multiple Shipping Services

Sellers who have *not* opted in to the Global Shipping Program can specify up to four domestic and five international shipping services for a listing.

Sellers who are participating in the Global Shipping Program can specify up to four domestic and four international shipping services, plus Global Shipping. This is the case regardless of whether or not the Global Shipping Program is actually offered for the listing.

If the Global Shipping Program is offered for a listing, it takes precedence, and becomes the only available international shipping option offered to any Global Shipping-eligible buyer who views that listing. However, if the buyer or the item is determined by eBay or the international shipping provider to be ineligible for the Global Shipping Program, the other international shipping options will automatically be offered for the listing instead.

For more information about shipping services, see “[Specifying Shipping Services](#)” on page 327.

Shipping Types

When offering shipping services, the seller specifies the type of shipping (which determines which tags are used and affects how costs are established for shipping of the item). The options for `ShippingType` are these:

- `Flat`
- `Calculated`
- `Freight` and `FreightFlat`
- `CalculatedDomesticFlatInternational`
- `FlatDomesticCalculatedInternational`

Note: The `CalculatedDomesticFlatInternational` and `FlatDomesticCalculatedInternational` types can be used to specify separate shipping types for the domestic and international legs of a traditional international shipment. Certain limitations apply to using shipping types with the Global Shipping Program. See “[Global Shipping Types](#)” on page 316.

Flat

Flat rate shipping (or “flat shipping”) is when the seller specifies a fixed shipping rate that the buyer is to pay, regardless of package dimensions or weight, and regardless of distance. The seller can offer various shipping services to the buyer (for example, USPS Airmail Parcel Post and UPS Express Plus) and decides how much the buyer would have to pay for each service, regardless of the actual cost of such shipping to the seller. The seller can offer or require insurance at a value set by the seller. See “[Shipping Insurance](#)” on page 362. The seller can also tack on a surcharge in certain circumstances. See “[Shipping Surcharge](#)” on page 351.

Calculated

Calculated rate shipping (or “calculated shipping”) is when a seller offers shipping services and requests that eBay calculate the shipping cost according to such things as the package’s dimensions and weight and the buyer’s postal code. The seller can offer or require insurance, but the insurance cost is determined by the selected shipping service (and is not known until the final sale price is known for the item). The seller can also charge the buyer packaging/handling costs in addition to whatever the shipping service would charge. See “[Packaging and Handling Cost](#)” on page 350.

Freight

Freight shipping is the shipping of items that are heavier than can be shipped with standard flat or calculated rate services. See “[Specifying Freight Shipping](#)” on page 346.

Global Shipping Types

For the domestic leg of a Global Shipping Program shipment, sellers can list an item using a **ShippingType** of either [Calculated](#) or [Flat](#). Sellers do not control the shipping type used for the international leg, and they do not control the total shipping cost, as the international shipping provider is responsible for all aspects of the international leg of the shipment.

Buyers see the total cost of the domestic and international shipping components as a single amount, not displayed separately. Although sellers can’t predict what the total shipping cost will be, they can control their cost structure for the domestic leg using **ShippingType**.

Other **ShippingType** values aren’t available for the Global Shipping Program. For more information about shipping types, see “[Specifying Shipping Types and Costs](#)” on page 336.

Shipping Costs

Shipping costs might include any or none of the following:

- The cost of shipping the first item in a multi-quantity listing and the cost of shipping each additional item in such a listing—see flat shipping details in “[Mixing Shipping Types](#)” on page 336
- Shipping insurance—see “[Shipping Insurance](#)” on page 362
- Packaging and handling—see “[Packaging and Handling Cost](#)” on page 350
- Shipping surcharge for certain destinations—see “[Shipping Surcharge](#)” on page 351
- Taxes—see “[Taxes](#)” on page 351
- Import charge—the total amount of customs duties, taxes, and related costs for shipping items across an international border as part of a Global Shipping Program shipment.

A key factor in determining applicable shipping costs (when listing) or parsing results is the shipping type.

Allocation of Costs with the Global Shipping Program

For Global Shipping, any costs that are associated with the international leg of the shipment are managed by the international shipping provider, who calculates them and incorporates them into either the shipping charge or the import charge. The import charge is determined by the

international shipping provider and presented to buyers on the listing page for an item, based on the buyer's location. This quote is estimated on the View Item Page (VIP) and guaranteed at checkout.

The seller can impose charges for packaging and handling costs for only the domestic leg of the shipment. The seller determines these costs the same way as for a purely domestic shipment, then must incorporate them into either the item price or the shipping charge, which along with the import charge are the only costs visible to the buyer per item.

Note: A buyer and seller can mutually agree to cancel part of a Global Shipping order after checkout and before it ships, and the buyer will receive a refund per the seller's refund policy.

However, due to the timing of this event, the buyer will not be refunded any shipping or import charges from the international leg of the shipment. It is recommended that buyers be made aware of this possibility in advance. When it occurs, they can accept the refund offered, or they can cancel the entire order and create a new order without the unwanted items.

Participation in the Global Shipping Program

The minimum compatibility level for the Global Shipping Program is 787.

By opting in to the Global Shipping Program, a seller can create listings for qualified items in which the international shipping option is Global Shipping (presented to buyers as "International Priority Shipping"). When a buyer in another country purchases a Global Shipping-enabled item, the buyer pays a single amount that covers the sum of the total item price, total shipping and handling for the item, and total import charges (i.e. customs duty and taxes) to transport the package into the buyer's country.

The seller ships the package to the domestic address of the international shipping provider, which then takes complete responsibility for transporting the package to the buyer, including selecting a delivery carrier, generating customs paperwork, and handling other details involved in an international shipment.

The seller's responsibility for shipping remains the same as for any domestic shipment, and ends when the international shipping provider takes charge of the shipment.

Currently, sellers may offer shipping via the Global Shipping Program in their listings, subject to the following qualifications:

- The seller must be opted in to the Global Shipping Program.
- Payment for the order must be via PayPal.
- The total value of the order (including shipping but not import charge) cannot be more than USD 2,500.00 (USD 1,600.00 for buyers in Canada).
- The package must not weigh more than 66 lbs.
- The package must not exceed 66 inches in length.
- The package must not exceed the maximum dimensions of 118 inches.
Dimensions = length + 2 * (width + height). Length is the package's longest side.

Item Eligibility

Items must also meet certain eligibility requirements before they can be listed with the Global Shipping Program:

- The listed item's location must be within the United States.
- The item must be listed on eBay.com.
- The item must not be illegal, hazardous, subject to export restrictions, [prohibited by eBay](#) or prohibited by the Global Shipping Program. See the [Global Shipping Program seller terms and conditions](#) for complete details.
- The item must be in a Global Shipping-eligible item category.

The current implementation of the Global Shipping Program applies to a subset of eBay's item categories. This subset includes a large number of categories, and over time, more categories will be included in the program.

The eligibility of a given item in a participating category to be shipped via the Global Shipping Program is automatically determined by the international shipping provider as soon as the seller selects Global Shipping as the default international shipping option for that item. If the item doesn't qualify, the seller will see a message requesting that a different international shipping option be selected.

You can determine whether a given item category is eligible for Global Shipping on the current eBay site. See "[Retrieving Global Shipping Program Status](#)" on page 319.

If a listing has both Global Shipping and another international shipping option enabled, and the item listed is determined by eBay or the international shipping provider to be ineligible for the Global Shipping Program, the Global Shipping option will be silently removed from the listing, and the remaining shipping option will automatically be offered in the normal way.

Payment Method

To be eligible for Global Shipping, a listing must specify [PayPal](#) as the sole payment method using [Item.PaymentMethods](#). If any additional payment method is specified in the listing, the listing won't be eligible for the Global Shipping Program.

Buyer Eligibility

When a buyer views the listing of a Global Shipping-enabled item, eBay compares the buyer's primary shipping address to the destination shipping regions and countries excluded by the seller for the listing, as well as to the international shipping provider's current import/export restrictions, to determine if the buyer can be offered shipping via Global Shipping for the item. If the buyer qualifies for the Global Shipping Program, the listing page displays it as the international shipping option and displays the estimated total cost of shipping, including import charges.

Duplicate Global Shipping Data Fields

Processing Global Shipping-enabled orders requires two complementary instances of certain transaction data to be returned (typically by one of the Get class of calls), reflecting the different requirements of the participants. This division of data varies depending on the type of information involved, as follows:

Tracking Information

The API call returns two instances of the `ShippingDetails.ShipmentTrackingDetails` container. In one instance, the `ShippingCarrierUsed` field specifies a domestic carrier and the `ShipmentTrackingNumber` field provides the domestic leg tracking information. In the other instance, the `ShippingCarrierUsed` field specifies `GlobalShipping_MultiCarrier` (the international shipping provider), and the `ShipmentTrackingNumber` field provides the end-to-end tracking information. The buyer sees only the latter.

Refund Amount

The API call returns two instances of the `ExternalTransaction.PaymentOrRefundAmount` field. The two values are the amount that the seller is responsible for refunding, and the amount that the international shipping provider is responsible for refunding. The buyer sees the total of the two refund amounts.

Additional Resources

See also:

Shipping Services

<http://pages.ebay.com/sellerinformation/shipping/chooseservice.html>

Frequently Asked Questions About Calculated Shipping Rates

<http://pages.ebay.com/help/sell/shipping-faqs.html>

Working with Shipping in API Calls

Shipping is not a distinct eBay API. Shipping functionality is achieved through the use of various containers, fields, types and values within eBay's [Trading API](#) and [Shopping API](#).

- “Setting Versus Retrieving Shipping Details” on page 319
- “Specifying Global Shipping in an Item Listing” on page 320
- “Specifying Global Shipping-Related Item Attributes” on page 320
- “Revising Shipping Details” on page 321
- “Using Global Shipping Information After Checkout” on page 322
- “Retrieving a Seller’s Global Shipping Records” on page 323

Setting Versus Retrieving Shipping Details

The `Item` container is where you specify shipping details when adding or revising a listing via the Add Item family of calls. See “[Managing Item Listings](#)” on page 375.

Retrieving shipping details is done via such calls as `GetItem`, `GetItemTransactions`, `GetItemShipping` and `GetOrders`. See “[Retrieving and Modifying Data About Sold Items](#)” on page 400.

Retrieving Global Shipping Program Status

To determine whether Global Shipping is available on the current eBay site, call `GetCategoryFeatures` with a `FeatureID` value of `GlobalShippingEnabled`. If `SiteDefaults.GlobalShippingEnabled = true`, Global Shipping is available. You can also provide a category name in the `CategoryID` field to

determine if the specified category is eligible for Global Shipping (`Category.GlobalShippingEnabled = true`). For more information, see the [GetCategoryFeatures call reference](#).

To determine if the current user has opted in to Global Shipping, call `GetUserPreferences` with a `ShowGlobalShippingProgramPreference` value of `true`. If `OfferGlobalShippingProgramPreference = true`, the caller has opted in to the Global Shipping Program. For more information, see the [GetUserPreferences call reference](#).

Specifying Global Shipping in an Item Listing

Opting in to the Global Shipping Program through the web UI causes a seller's shipping preferences to include the option to offer international shipping via the Global Shipping Program. This option's default value is `Yes`. The Global Shipping service `InternationalPriorityShipping` becomes the default first international shipping option for any new item listings created.

Once a seller has opted in to the Global Shipping Program, it becomes an option the seller can offer when listing an item. To list an item with Global Shipping as an international shipping option, use any API listing call that contains the `ShippingDetails.GlobalShipping` field, and set its value to `True`. This is the only step needed to configure the international portion of shipping for the item, because the international shipping provider handles the details of that leg. The default Global Shipping Program service, `InternationalPriorityShipping`, is automatically set for the listing.

Listing calls with the `ShippingDetails.GlobalShipping` input field include `AddFixedPricelItem`, `AddItem`, `AddItems`, `RelistFixedPricelItem`, `RelistItem`, `ReviseFixedPricelItem` and `ReviselItem`.

To confirm that a listing offers Global Shipping, the Trading API's `GetItem` provides the `ShippingDetails.GlobalShipping` output field, and the Shopping API's `GetSingleItem` and `GetMultipleItems` calls provide the `Item.GlobalShipping` output field.

The listed item must also meet the Global Shipping Program eligibility requirements. See ["Participation in the Global Shipping Program"](#) on page 317.

If the seller uses `ShippingDetails.InternationalShippingServiceOption.ShippingService` to specify (up to four) additional international shipping options for the same listing, Global Shipping will automatically take precedence, and International Priority Shipping will be the sole international shipping service offered to any Global Shipping-eligible buyer who views that listing. However, if the buyer or the item is determined by eBay or the international shipping provider to be ineligible for the Global Shipping Program, the other international shipping options will automatically be offered for the listing instead.

Sellers must also configure shipping settings for the domestic leg of the shipment. All of the normal domestic shipping settings are available for that leg, and sellers can use the same techniques to configure those settings as for any listing with domestic shipping.

Specifying Global Shipping-Related Item Attributes

With the Global Shipping Program, the international shipping provider makes use of two item characteristics in determining customs duty, taxes, and whether an item can be shipped to a particular country. Sellers can provide either or both of these two characteristics when listing an item. They're stored as item attributes using the `Item.ItemSpecifics` container that's available with most Trading API calls that create or update item listings, such as `AddItem` or `ReviseFixedPricelItem`.

Each instance of `Item.ItemSpecifics.NameValueList` contains a name/value pair. For the Global Shipping attributes, the names are:

- `<Name>Country Of Manufacture</Name>`
The `<Value>` provided for Country Of Manufacture should identify the country in which more than 50% of the item's value was created or added.
- `<Name>Harmonized System Code</Name>`
The Harmonized System Code is an international standard item classification used to determine customs duty, taxes, and certain restrictions on international shipments. The `<Value>` provided for Harmonized System Code should specify the first 6 digits of the official code corresponding to the item, as determined by the [U.S. Official Harmonized Tariff Schedule](#). Use only numeric digits for this value, and omit any decimal point or other separator.

These two attributes are currently optional. For more information about `ItemSpecifics`, see “[Working with Custom Item Specifics](#)” on page 481.

Multiple Packages

If the shipment for an order is split into two or more packages, the `Country Of Manufacture` value becomes mandatory. However, in this case it's not provided in `ItemSpecifics`, but rather in a dedicated field, `CountryOfOrigin`, using the `CompleteSale` call. For more information, see the [CompleteSale call reference](#).

Revising Shipping Details

If your goal is to revise shipping details for an existing listing, you should be aware that the approach for revising the listing differs from the typical approach for revising a list.

To avoid loss of shipping details, do not omit any tag when revising shipping details, even if its value does not change. Omitting shipping fields while revising an item will remove that detail from the listing.

For example, if you specify `ShippingDetails` when you revise or relist an item but you omit its child field `InternationalShippingServiceOption`, eBay will drop any international shipping services from the listing.

Note: One exception to this behavior is the boolean `ShippingDetails.GlobalShipping` field. If you omit this field when revising or relisting an item listing, its value will be unchanged. A listing that offers Global Shipping will continue to offer Global Shipping unless you explicitly include the `GlobalShipping` field with a value of `false`.

To find out the shipping fields present in the current listing, call `GetItem` and look through the `Item.ShippingDetails` fields in the response.

For multi-quantity Fixed Price listings, a seller can revise all shipping details of the listing (except for sales tax and for shipping type of Freight) for all unsold items. This applies to both domestic and international shipping. Checkout is not affected for those who bought items prior to the seller's shipping changes—the shipping details that were in effect at the time of purchase are used for that buyer at the time of checkout.

See also “[Item Changes Allowed During Revisions](#)” on page 382.

Using Global Shipping Information After Checkout

Sellers can use the [GetItemTransactions](#), [GetOrders](#), [GetOrderTransactions](#) or [GetSellerTransactions](#) call to retrieve a variety of information about a recent sale. Global Shipping Program information is returned in containers and fields in one of the following paths:

- [GetItemTransactions.TransactionArray.Transaction](#)
- [GetOrders.OrderArray.Order](#)
- [GetOrderTransactions.OrderArray.Order](#)
- [GetSellerTransactions.TransactionArray.Transaction](#)

The structure of the Global Shipping-specific containers and fields is as follows:

Example 17-1 Global Shipping Containers and Fields

```
<IsMultilegShipping>true</IsMultilegShipping>
<MultiLegShippingDetails>
    <SellerShipmentToLogisticsProvider>
        <ShippingServiceDetails>
            <ShippingService> </ShippingService>
            <TotalShippingCost currencyID="USD"> </TotalShippingCost>
        </ShippingServiceDetails>
        <ShippingTimeMin> </ShippingTimeMin>
        <ShippingTimeMax> </ShippingTimeMax>
        <ShipToAddress>
            ...
            <ReferenceID> </ReferenceID>
            ...
        </ShipToAddress>
    </SellerShipmentToLogisticsProvider>
</MultiLegShippingDetails>
```

To determine if the shipment for an order or transaction uses the Global Shipping Program, examine the boolean [IsMultilegShipping](#) field. If [IsMultilegShipping](#) is true, the shipment uses the Global Shipping Program.

To determine the seller-specified shipping service for the domestic leg, examine the [ShippingService](#) field. To determine the seller-specified shipping cost for the domestic leg, examine the [TotalShippingCost](#) field. To determine the seller-specified minimum and maximum number of shipping days for the domestic leg, examine the [ShippingTimeMin](#) and [ShippingTimeMax](#) fields.

Addressing a Global Shipping Program Shipment

The buyer’s shipping address is returned in the [ShippingAddress](#) container in the Get family of transactions. However, sellers should *not* ship to this address; instead, they should ship the order to the international shipping provider’s domestic depot. The international shipping provider handles the subsequent leg of the shipment.

When addressing a package for the first, domestic leg of a Global Shipping Program shipment to the international shipping provider, sellers obtain the addressing information using the

[GetItemTransactions](#), [GetOrders](#), [GetOrderTransactions](#) or [GetSellerTransactions](#) call. In each of these calls, the output structure that contains the domestic address is [MultiLegShippingDetails.SellerShipmentToLogisticsProvider.ShipToAddress](#).

This is the domestic address of the international shipping provider depot. It includes the [ReferenceID](#) field, the value of which must be included on the package above the street address. The name of the addressee is the buyer name as shown by the [Name](#) field.

For example:

Casey P. Buyer
Reference #1234567890123456
1850 Airport Exchange Blvd, Suite 400
Erlanger KY 41018
United States

The international shipping provider uses the [ReferenceID](#) value to retrieve the relevant details about the buyer and the order, so the shipment can be completed.

The shipping and import charges are calculated for each Global Shipping Program order, to be displayed to buyers before checkout. Because those calculations are based on sending the order as a single shipment, sellers are expected to send a single shipment.

Note: To ensure fast, smooth shipping and customs processing, sellers are strongly encouraged to ship all items belonging to an order in a single shipment. However, it might occasionally be physically impossible for a seller to send the entire order in a single shipment. If an order is sent as multiple shipments, the overall cost of shipping and import charges might increase, but sellers will not be charged for the extra cost. If in the future the use of multiple shipments becomes more prevalent, this policy is subject to change, and sellers could be held liable for the increased shipping costs.

Retrieving a Seller's Global Shipping Records

Several Trading API calls provide information about a seller's activity involving the Global Shipping Program. Following are the relevant fields returned:

GetMyeBaySelling

This call returns information about the current status of items that the seller has listed or sold under the Global Shipping Program.

- [ShippingDetails.GlobalShipping](#) indicates whether an item was listed with the Global Shipping Program as an international shipping option. This field is returned under the [ActiveList](#), [BidList](#), [DeletedFromSoldList](#), [DeletedFromUnsoldList](#), [ScheduledList](#), [SoldList](#) and [UnsoldList](#) containers.
- [Buyer.BuyerInfo.ShippingAddress.PostalCode](#) contains the postal code of the item's buyer. This field is returned in two locations under the [DeletedFromSoldList](#) container and in two locations under the [SoldList](#) container.
- [Transaction.IsMultiLegShipping](#) indicates whether an item was shipped using the Global Shipping Program. This field is returned in two locations under the [DeletedFromSoldList](#) container and in one location under the [SoldList](#) container.
- [ShippingDetails.ShippingServiceOptions.ShippingServiceCost](#) contains the cost of the domestic leg (to the international shipping provider's warehouse) of the item shipment. This field is

returned in two locations under the **DeletedFromSoldList** container and in two locations under the **SoldList** container.

- For orders awaiting payment, **TotalPrice** contains the purchase price of the items; for paid orders, **TotalPrice** contains the purchase price of the items plus the cost of domestic shipping to the international shipping provider's warehouse. Under **SoldList.OrderTransactionArray.OrderTransaction**, this amount appears:
 - in **Transaction.TotalPrice** for single-transaction orders, or
 - in **Order.TransactionArray.Transaction.TotalPrice** for multiple-transaction orders.
- For more information about these fields, see the [GetMyeBaySelling call reference](#).

GetSellerList

This call returns information about items that the seller has listed under the Global Shipping Program.

- **ShippingDetails.GlobalShipping** indicates whether an item was listed with the Global Shipping Program as an international shipping option.
- **SellingStatus.HighBidder.BuyerInfo.ShippingAddress** contains the **Country** and **PostalCode** of the item's buyer.

For more information about these fields, see the [GetSellerList call reference](#).

Specifying Locations to Where You Ship

["Using Shipping Rate Tables for Finer Control of Shipping Location Charges" on page 325](#)
["ShipToLocation\(s\)" on page 325](#)
["Excluding Locations" on page 326](#)
["US Listings and Package Size Limitations" on page 326](#)
["Specifying Shipping Locations with the Global Shipping Program" on page 327](#)

When listing, revising, or relisting items with the Add/Revise/Relist family of API calls, the optional **Item.ShipToLocations** field lets you specify the regions or countries to where you will ship items. An alternative method is to list the regions or countries you will *not* ship to by using the **ExcludeShipToLocation** field.

Pass either **ShippingLocationDetails** or **ExcludeShippingLocationDetails** as an argument to **DetailName** in a call to [GeteBayDetails](#) to get a list of the locations and countries that you can include or exclude, respectively, as destinations to which you will ship. For more information, see the [GeteBayDetails call reference](#).

If a buyer's primary ship-to location is a location that you have listed as an excluded ship-to location (or if the buyer does not have a primary ship-to location), they will receive an error message if they attempt to buy or place a bid on your item.

Using Shipping Rate Tables for Finer Control of Shipping Location Charges

On the DE, UK and US sites, sellers can configure flat shipping rate tables in My eBay Shipping Preferences. These tables can then be applied to individual listings using the Trading API. Shipping rate tables enable sellers to tailor the flat shipping rates offered for an item to fit the shipping destination. They can specify a *base* rate for the seller's country or an international region, then specify different rates or surcharges for shipping to the more distant or inconvenient areas within the country or to specific countries within the region.

For more information about configuring and invoking flat shipping rate tables, see "[Using Shipping Rate Tables](#)" on page 338.

ShipToLocation(s)

There is a **ShipToLocations** at the **Item** level of the **AddItem** family of calls and a **ShipToLocation** inside **InternationalShippingServiceOptions**. The purpose of this section is to illustrate the difference between the two.

(Use **GeteBayDetails** to determine which locations are valid per site. These values are string equivalents of values found in **ShippingRegionCodeType** and **CountryCodeType**.)

Item-Level ShipToLocations

As input, the **Item**-level **ShipToLocations** enables a seller to identify one or more locations to which he is willing to ship the item in addition to the country of his own site. There is no connection to whether the seller is offering any international shipping services.

As output, the **Item**-level **ShipToLocations** reflects the consolidated list of both **Item**-level **ShipToLocations** and service-level **ShipToLocation** values.

For example, if a seller specifies several **Item**-level **ShipToLocations** and one of those is **Worldwide**, **GetItem** returns **Worldwide** as the only **Item**-level **ShipToLocation**. If the combination of all **Item**-level **ShipToLocations** and service-level **ShipToLocation** values amounts to worldwide, **GetItem** returns **Worldwide** as the only **Item**-level **ShipToLocation**.

InternationalShippingService-Level ShipToLocation

As input, the **ShippingDetails.InternationalShippingServiceOption.ShipToLocation** field represents the locations to which the seller is willing to ship the item with that particular international shipping service. For each international shipping service the seller offers, the seller must provide at least one **ShipToLocation**. There is no connection to whether **ShipToLocations** has been specified at the **Item** level.

Note: This applies to the Global Shipping **InternationalPriorityShipping** service, subject to exclusions made by the international shipping provider.

A seller wishes to offer a couple of international shipping services as options to the buyer. At the same time, the seller does not want to specify where in the world he is willing to ship. Let's say he

determines that shipping service XYZ only ships to Canada and Europe and that shipping service ABC offers shipping to Canada at a lower cost than shipping service XYZ:

- The seller does not want to say where he's willing to ship, so he does nothing with **Item**-level **ShipToLocations**.
- The seller offers two international shipping services: service ABC and XYZ.
- Since ABC offers a price break for Canada, he specifies **CA** as the **ShipToLocation** corresponding to service ABC.
- He specifies **CA** and **Europe** as **ShipToLocation** for XYZ. (Why not? Perhaps some buyer prefers the services of XYZ to Canada, even though they are more expensive than ABC.)
- Because of the API requirement that at least one domestic shipping service be offered if any international shipping services are offered, the seller picks some domestic shipping service and then lists the item.
- **GetItem** returns **CA** and **Europe** as the **Item**-level **ShipToLocations**, since the **Item**-level **ShipToLocations** reflects the consolidated list of both **Item**-level **ShipToLocations** and service-level **ShipToLocation** values. (The service-level **ShipToLocation** values remain unchanged: whatever was input in the **InternationalShippingServiceOptions** container is seen in the output.)

If the buyer wants the item shipped to Mexico (a location not offered by the seller), perhaps the buyer can work out a deal with the seller via messaging/email/Skype.

Excluding Locations

The **ShipToLocations** and **ExcludeShipToLocation** fields work together. You can list a region (such as "Africa") in the **ShipToLocations** field, then exclude specific countries (such as "Chad") within that region with **ExcludeShipToLocation** fields. In addition, if your **ShipToLocations** is **Worldwide**, you can specify regions and countries that you wish to exclude from your shipping destinations with **ExcludeShipToLocation**.

You can use the **Exclude shipping locations** page in My eBay Shipping Preferences to set up a default set of locations where you will not ship items. Once set up, your excluded shipping locations will be set for all future listings, and if the **Apply to all current live listings** option is selected, all current listings. If you want to override the default list for a particular item, you can do so using the **ShipToLocations** and/or **ExcludeShipToLocation** fields as needed.

US Listings and Package Size Limitations

Not all combinations of shipping method and package size are allowed by all domestic and international shipping services for US listings. For details on **Package Sizes**, see the following eBay website page:

<http://pages.ebay.com/help/sell/ship-calc-package.html>

Specifying Shipping Locations with the Global Shipping Program

Sellers don't need to specify their own country as a location they will ship to at listing time, as eBay automatically sets the listing site's country as a `ShipToLocation` value. However, sellers can include the `ShippingService` field in the domestic shipping container to specify that they *do not* ship domestically at all (for example by specifying local pickup, buyer responsible for shipping, or pickup for eBay Motors).

The international shipping provider specifies the full set of available international shipping locations for a Global Shipping-enabled item. The criteria that determine which countries are eligible for the Global Shipping Program are subject to political, logistical, regulatory and cost considerations. However, the seller can exclude any countries from the set of eligible Global ShippingGlobal Shipping countries at the account level, or as part of an item listing.

Once a seller has opted in to the Global Shipping Program, it becomes the default shipping method for all Global Shipping-eligible destinations except those specifically excluded by the seller.

Specifying Shipping Services

- “Required Domestic Shipping Service” on page 327
- “Handling Time and Estimated Delivery Time” on page 329
- “Enabling Get It Fast” on page 331
- “Deprecated Shipping Services” on page 334

A seller can specify up to four domestic shipping services and up to five international shipping services (or four international shipping services plus Global Shipping if the seller is opted in to the Global Shipping Program). To get basic facts about each available shipping service, call `GetEBayDetails` with `DetailName` set to `ShippingServiceDetails`.

Note: The Global Shipping Program is represented by the `InternationalPriorityShipping` service. However, this service is not specified directly through the API. Instead, you set the value of the `ShippingDetails.GlobalShipping` field to `true` for listing, and the international shipping service is automatically set to `InternationalPriorityShipping`. For more information, see “Working with Shipping in API Calls” on page 319.

For more information about shipping services, see “[Shipping Carriers and Services](#)” on page 314.

Note: For the domestic leg of a Global Shipping Program shipment, sellers can use any carrier and service that is available to them for U.S. domestic shipping.

Required Domestic Shipping Service

Although requirements for specified shipping vary by site, sellers are strongly encouraged to add specified shipping to all listings. If items in a category require specified shipping, then `GetCategoryFeatures` returns `true` for `Category.ShippingTermsRequired`.

On the Germany (DE) and Spain (ES) sites, users must specify at least one domestic shipping service.

On the US site, sellers must specify at least one domestic shipping service, and must specify if the shipping is flat or calculated. The exception is freight shipping, and if freight shipping is specified, there is not a requirement to specify the cost.

In most cases, a seller specifies one or more shipping services for flat or calculated shipping (see “[Specifying Flat Rate Shipping](#)” on page 336, “[Specifying Calculated Shipping](#)” on page 341).

Sometimes, though, a seller does not wish to (and, in certain cases, cannot) select a shipping service for a particular shipping carrier. Consider these scenarios:

- “I will not ship the item. You must arrange for pickup of the item.”
- “Shipping is free for you!”
- “I will ship via FedEx. The cost to you (beyond the cost of the item) will be the actual FedEx charge plus \$2 for my trouble.”
- “If you are the winning bidder, contact me and tell me how you want it shipped, and we'll work out the cost.”
- “This item requires freight shipping. I will pick the freight shipping carrier and work out the details with you once I know your location.”

Specifying Local Pickup

Local pickup is an interesting case: there is a shipping service enumeration named `LocalDelivery` in the same list of enumerations as the `USPSPriority` service. Obviously, the seller pays no shipping carrier to handle local pickup. But the eBay API mechanism is to use the “shipping service” named `LocalDelivery`.

If a seller does not intend to ship an item, set `Item.ShippingDetails.ShippingServiceOptions.ShippingService` to `LocalDelivery`.

Note: There is slightly different behavior on the eBay website. First time sellers listing an item through eBay website must provide at least one domestic shipping service.

Specifying Free Shipping

(For the eBay US website only.) When `ShippingServiceOption.FreeShipping` is set to true, the cost of shipping is zero for the first domestic shipping service (for flat or calculated shipping).

If `InsuranceOption` is `Required` and `FreeShipping` is `true`, eBay changes the insurance cost to 0 for that shipping service. But if `InsuranceOption` is `Optional`, the cost of shipping insurance remains, for it is up to the buyer whether to buy shipping insurance, regardless of whether the seller specified free shipping.

This applies to the `AddItem` family of calls. This element is also returned by `GetItem` and `GetItemShipping` if the first domestic shipping service was marked as `FreeShipping`.

Specifying Other Shipping Solutions

In rare cases, no shipping service in eBay’s `ShippingServiceCodeType` matches what the seller wants to do. The seller can add information to the item description (`Item.Description`) to explain the shipping situation to bidders.

Handling Time and Estimated Delivery Time

The handling time (also known as the dispatch time) refers to the following time period:

- From the time seller has received cleared payment,
- To the time the seller has delivered the package to the shipping carrier

For most item categories, sellers are required to use **DispatchTimeMax** (in add, relist and revise calls) to specify an item's handling time in days (a positive integer) for domestic delivery.

Note: For the Global Shipping Program, the information here about handling time is applicable. eBay calculates the total estimated delivery time by adding the seller's stated handling time (**DispatchTimeMax**) to the seller's estimate of the delivery time for the shipping service used for the domestic leg, plus the international shipping provider's estimate for the international leg.

DispatchTimeMax does not apply when there is no shipping, or when using local pickup only or freight shipping. For example, when **ShippingService**=**Pickup** or **ShipToLocations**=**None**, then **DispatchTimeMax** is not required (or it can be empty).

DispatchTimeMax also does not apply to certain listing categories such as Vehicles, Real Estate, Specialty Services and Classified Ads listings. See **HandlingTimeEnabled** in **GetCategoryFeatures** to determine exactly which categories require this field.

Note: **AddItem** requires that you specify a **DispatchTimeMax** for items listed in certain categories. Calls will fail if the required dispatch time is not specified. See the *eBay Developer Newsletter* and the blog for details of this policy.

For a list of allowed **DispatchTimeMax** values on each eBay site, use **DispatchTimeMaxDetails** in **GeteBayDetails**. Typical values are 0, 1, 2, 3, 4, 5, 10, 15, or 20, but they can vary by site and may change over time.

Same Day Handling

You can specify same day handling by setting **DispatchTimeMax** to 0. This means that the seller commits to shipping the packaged order by the end of the business day.

Note: Same day handling is available on the eBay Sandbox starting in release 817. It becomes available on the production site as of Seller Release 13.1.

The meaning of "business day" in this context depends on the locale of the listing site and the time the cleared payment is received. Each eBay site has a default *order cut off time*. For orders placed (and cleared payment received) before the local order cut off time, the order must be shipped by the end of the current day. For orders completed at or after the order cut off time, the order must be shipped by the end of the following day (excluding weekends and local holidays).

For example, the default order cut off time on the eBay US site is 2:00 pm PT. An order placed on that site on a Friday by 1:59 pm PT must be shipped by the end of business that day. However, an order placed on a Friday on or after 2:00 pm PT must be shipped by the end of business the following Monday (assuming Monday is not a local holiday), and the estimated delivery is calculated from that date.

Note: If a same day shipping carrier is selected, and the carrier delivers on one or both weekend days, sellers on the eBay US site are assumed to be open for business on the same days, and those days will be used when calculating total shipping time.

The following table lists default order cut off times for each active eBay site:

Table 17-2 Default Order Cut Off Times by eBay Site

Site ID	Site Code	Country Detail	Default Order Cut Off Time
0	US	US	2 PM PT (UTC/GMT -7 hours; DST: +1 hour)
2	CA	Canada	2 PM ET (UTC/GMT -5 hours; DST: +1 hour)
210	CAF.R	Canada (French)	2 PM ET (UTC/GMT -5 hours; DST: +1 hour)
3	UK	United Kingdom	2 PM BST (No UTC/GMT offset; DST: +1 hour)
15	AU	Australia	2 PM AET (UTC/GMT +10 hours; DST: +1 hour)
16	AT	Austria	2 PM CET (UTC/GMT +1 hour; DST: +1 hour)
23	BE_FR	Belgium (French)	2 PM CET (UTC/GMT +1 hour; DST: +1 hour)
71	FR	France	2 PM CET (UTC/GMT +1 hour; DST: +1 hour)
77	DE	Germany	2 PM CET (UTC/GMT +1 hour; DST: +1 hour)
100	MOTORS	US Motors	2 PM PT (UTC/GMT -7 hours; DST: +1 hour)
101	IT	Italy	2 PM CET (UTC/GMT +1 hour; DST: +1 hour)
123	BE_NL	Belgium (Dutch)	2 PM CET (UTC/GMT +1 hour; DST: +1 hour)
146	NL	Netherlands	2 PM CET (UTC/GMT +1 hour; DST: +1 hour)
186	ES	Spain	2 PM CET (UTC/GMT +1 hour; DST: +1 hour)
193	CH	Switzerland	2 PM CET (UTC/GMT +1 hour; DST: +1 hour)
201	HK	Hong Kong	2 PM HKT (UTC/GMT +8 hours; No DST)
205	IE	Ireland	2 PM IST (No UTC/GMT offset; DST: +1 hour)
207	MY	Malaysia	2 PM MYT (UTC/GMT +8 hours; No DST)
211	PH	Philippines	2 PM PHT (UTC/GMT +8 hours; No DST)
212	PL	Poland	2 PM CET (UTC/GMT +1 hour; DST: +1 hour)
216	SG	Singapore	2 PM SGT (UTC/GMT +8 hours; No DST)

You can override the default order cut off time for individual users. Use the [SetUserPreferences](#) call and specify a new cut off time for the authenticated user in

[DispatchCutOffTimePreference.CutOffTime](#). Provide the time value in Universal Coordinated Time (UTC/GMT), accounting for any difference in daylight saving time between GMT and the seller's time zone if necessary.

You can inspect the currently set order cut off time for the authenticated user by calling [GetUserPreferences](#) and setting [ShowDispatchCutoffTimePreferences](#) to `true`. The response returns the cut off time in [DispatchCutOffTimePreference.CutOffTime](#).

Combining Handling and Delivery Time

Optionally, you can estimate an item's total delivery time and display it in your application. For example, you could display "Estimated delivery: 5 to 7 days" for an item. To do this, you need to know the seller's handling time as well as the estimated shipping (transit) time for the shipping service. [GeteBayDetails](#) includes an estimated transit time range for most shipping services, specified by [ShippingTimeMin](#) and [ShippingTimeMax](#).

To present a total estimated delivery time to your users, use the following calculations:

Low end of delivery time range = **DispatchTimeMax + ShippingTimeMin**

High end of delivery time range = **DispatchTimeMax + ShippingTimeMax + buffer**

The buffer can be set by the seller, or it can be determined by your application. For example, during the holiday season, the shipping carrier may recommend that you add 1 day as a buffer. (eBay does not provide a recommended buffer.)

Exception Handling Time

If **GeteBayDetails** returns a **true** value for **DispatchTimeMaxDetails.ExtendedHandling**, the seller has specified a default handling time of 4 business days or more (an *exception handling time*) in preferences. You might want to warn the seller that long handling times might discourage potential buyers.

Enabling Get It Fast

The **ShippingServiceDetails** container includes **ExpeditedService**, **ShippingTimeMax** and **ShippingTimeMin** elements. These play a role in Get It Fast listings.

Many buyers want or need to receive an item as soon as possible after purchase (for example, last minute shopping for the holidays). Such buyers benefit from knowing if purchasing an item from one seller would get the item into the buyer's hands faster than if purchased from another seller.

Buyers can search for Get It Fast listings, listings that have been opted into the Get It Fast feature by sellers. By opting a listing into the Get It Fast feature, a seller commits to delivering the package to the buyer-selected shipping service within one working day after receiving payment, even if that service is not an expedited shipping service.

Requirements for Get It Fast

The following requirements must be met for a seller to opt a listing into the Get It Fast feature:

- The site must support Get It Fast (see “[Miscellaneous Item Field Differences](#)” on page 600).
- The seller must offer at least one domestic expedited shipping service.

An expedited service is one that has been defined by the site to have a guaranteed fast delivery time. You can determine if a shipping service is an expedited service by calling [GeteBayDetails](#) with [DetailName](#) set to [ShippingServiceDetails](#) and inspecting the value of [ExpeditedService](#).

- [GetItFast](#) must be set to [true](#) when listing the item.
- The listing must be Fixed Price or have a Buy It Now price.

Once a bid is made on an auction listing for which there is a Buy It Now price, the listing ceases to be a Get It Fast listing.

- [DispatchTimeMax](#) must be set to [0](#) or [1](#) when listing the item.

[DispatchTimeMax](#) specifies the maximum number of business days ([0](#), [1](#), [2](#), [3](#), [4](#), [5](#), [10](#), [15](#), or [20](#)) the seller commits to for shipping an item to buyers after receiving a cleared payment. This does not include the carrier’s shipping time (in transit).

Sellers may also specify [DispatchTimeMax](#) without opting a listing into the Get It Fast feature.

Get It Fast Shipping Details

Get It Fast is for domestic shipping only (shipping within country of listing site). A buyer only sees Get It Fast as a feature if the seller opted the listing into Get It Fast (per the rules above) and if the buyer’s shipping address is within the same country as the listing site.

An expedited shipping service must be associated with the Get It Fast feature to be called a Get It Fast shipping service. Note that when at least one of three domestic services offered by the seller is identified as a Get It Fast service, all domestic services offered are to be treated by the seller as Get It Fast services: the seller’s commitment to ship within one day applies regardless of the service the buyer picks.

Note that a shipping service being “expedited” according to [GeteBayDetails](#) does not mean that the service is a Get It Fast service. The listing must have been intentionally opted into the Get It Fast feature for any expedited services to be Get It Fast services. Thus, [GetItem](#) returns the [GetItFast](#) boolean (indicating opt-in state) in addition to the [ExpeditedService](#) boolean for each domestic shipping service.

Each shipping service identified by [GeteBayDetails](#) as an [ExpeditedService](#) has an associated [ShippingTimeMin](#) and [ShippingTimeMax](#). The buyer benefits from knowing such information in deciding which shipping service to select.

Get It Fast Handling Details

Although sellers may use [DispatchTimeMax](#) to specify a domestic handling time without committing to participation in the Get It Fast program, [DispatchTimeMax](#) is required and must be set to 1 (1 business day) to qualify for Get It Fast. [DispatchTimeMax](#) is not limited to fixed price or Buy It Now listing formats, except when it is used with Get It Fast.

DispatchTimeMax will not be automatically added to Get It Fast listings active prior to the introduction of **DispatchTimeMax**. However, when a Get It Fast item is added, revised, or relisted without specifying **DispatchTimeMax**, the **DispatchTimeMax** is set to 1, the default value that qualifies for Get It Fast.

Whether **DispatchTimeMax** is set or not, the View Item page will display a handling time of 1 business day for Get It Fast items.

Related API Functions and Tags

The minimum compatibility level for Get It Fast is 353.

The following calls and fields enable Get It Fast:

- The **AddItem** family of functions (input):

Item.GetItFast specifies whether the item is to be opted into the Get It Fast feature

DispatchTimeMax specifies the handling time, in business days, for items to be shipped domestically (must be set to **1** to enable Get It Fast)

The following functions and tags can be used to get the status of the Get It Fast feature:

- **GetItem** output:

Item.GetItFast (whether the item was opted into Get It Fast)

For domestic services only: **ShippingDetails.ShippingServiceOptions**: **ExpeditedService**, **ShippingTimeMin**, **ShippingTimeMax**

- **GetOrders** output:

OrderArray.Order.ShippingDetails.GetItFast

For domestic services only: **ShippingDetails.ShippingServiceOptions**: **ExpeditedService**, **ShippingTimeMin**, **ShippingTimeMax**

- **GetSellerTransactions** output:

Item.GetItFast

Transaction.ShippingDetails.GetItFast

- **GetItemTransactions** output: Same as that of **GetSellerTransactions**, plus:

For domestic services only:

Transaction.ShippingDetails.ShippingServiceOptions: **ExpeditedService**, **ShippingTimeMin**, **ShippingTimeMax**

- **GetOrderTransactions**: See comments regarding **GetOrders** and **GetItemTransactions**.
- **GetSellerList**: **ItemArray.Item.GetItFast** (whether the item was opted into the Get It Fast feature)
- **GetItemShipping** output:

For domestic services only: **ShippingDetails.ShippingServiceOptions**: **ExpeditedService**, **ShippingTimeMin**, **ShippingTimeMax**

- **GeteBayDetails**: A **DetailName** value of **ShippingServiceDetails** returns an **ExpeditedService** value for each shipping service, indicating whether that service is an expedited shipping service that could qualify in the Get It Fast feature as a Get It Fast shipping service and, if there are such values, what are the minimum and maximum number of days quoted by the shipping service for delivery.

Site Help for Get It Fast

The following help is also available to users via links in the Shipping portion of the website:

Get It Fast programme

<http://pages.ebay.co.uk/help/sell/contextual/getitfast.html>

Deprecated Shipping Services

All valid shipping services for an eBay site will be indicated by a **ShippingServiceDetails.ValidForSellingFlow** flag in the **GeteBayDetails** output. If a shipping service returned in **GeteBayDetails** does not contain this flag, the shipping service is no longer available/valid, and the seller can not specify this shipping service as a shipping option for buyers.

API vs Web

Response to a user selecting a deprecated service is different in the API version of the Sell/Revise/Relist Your Item flows than it is in the eBay website version.

A new API listing that offers a to-be-deprecated shipping service will not be blocked; a warning will be returned that that particular shipping service will be deprecated.

A new API listing with an already-deprecated shipping service will not be blocked; that service will automatically be mapped by eBay to the service noted in **MappedToShippingServiceID**, and a warning will be returned about this action.

On the eBay website, the general behavior is to inform the buyer or seller and they must decide whether to keep the current service or go with a different service. If the service has been deprecated, the user either accepts the mapped service or selects some other service.

API Specifics

```
DeprecationDetails ( AnnouncementMessageType )
  AnnouncementStartTime ( dateTime )
  EventTime ( dateTime )
  MessageType ( None/Deprecation/Mapping/DeprecationAndMapping )
  MappedToShippingServiceID
```

In the **ShippingServiceDetails** container returned by **GeteBayDetails** for each shipping service, you find **DeprecationDetails** and **MappedToShippingServiceID** only if the shipping service will be or has been deprecated.

MappedToShippingServiceID identifies the numerical id of the service that is recommended as the replacement for the deprecated shipping service.

DeprecationDetails does not contain any message about deprecation. Rather, **DeprecationDetails** contains information about which messages are to be made available and when. It is part of a mechanism that enables eBay to decide, per country, for example, which messages to make available at what time.

EventTime is when the event will occur, which, in this context, is when the shipping service will be deprecated. **AnnouncementStartTime** is when messaging can begin about the upcoming **EventTime**.

MessageType identifies the kind of message that eBay wishes to make available for this particular event. Options are to have no messaging related to the event, to have messaging during the to-be-deprecated period (between **AnnouncementStartTime** and **EventTime**), to have messaging during the mapping period (any time after **EventTime**), and to have messaging in both the deprecation and mapping periods.

(At this time, **DeprecationDetails** can be returned empty. This happens when **GeteBayDetails.ValidForSellingFlow** is **false** for a shipping service and the shipping service is deprecated and there is no mapping for that deprecated shipping service. Interpret an empty **DeprecationDetails** as if it were a message that there is no mapping for that deprecated service, for eBay will not map the service – it will simply drop the shipping service and will not return a warning message.)

Example

You call [GeteBayDetails](#) and get this information as part of the [ShippingServiceDetails](#) container for a shipping service:

```
AnnouncementStartTime = June 1
EventTime = July 1
MessageType = DeprecationAndMapping
MappedToShippingServiceID = 123456
```

[MessageType](#) is set to [DeprecationAndMapping](#) which effectively uses both the [Deprecation](#) and the [Mapping](#) enumerations. Messages are to be issued during both the deprecation and the mapping periods.

If it is before June 1, no warning is returned because the announcement period has not started. Between June 1 and July 1, a warning is returned to those who create a new listing via API that includes the to-be-deprecated shipping service. After July 1, if someone lists via API with this deprecated service, eBay maps it to service 123456 and returns a warning that this has occurred.

Specifying Shipping Types and Costs

- ["Mixing Shipping Types" on page 336](#)
- ["Specifying Flat Rate Shipping" on page 336](#)
- ["Specifying Calculated Shipping" on page 341](#)
- ["Package Dimensions and Weight" on page 342](#)
- ["Specifying Freight Shipping" on page 346](#)
- ["UPS Rate Scales" on page 349](#)
- ["Search Results and Shipping Costs" on page 350](#)

Mixing Shipping Types

In offering multiple shipping services, sellers can mix different shipping types (for example, [Flat](#) or [Calculated](#)) depending on the circumstances, as follows:

All domestic and international shipping services offered by the seller must be the same shipping type, specified by one field, [Item.ShippingDetails.ShippingType](#). However, for eBay US and for Motors Parts and Accessories, a seller can select one shipping type for all domestic shipping services, and a different type for all international shipping services. The [ShippingType](#) values [CalculatedDomesticFlatInternational](#) and [FlatDomesticCalculatedInternational](#) exist for this reason. (See also [InternationalShippingServiceOptionsType](#) and [ShippingServiceOptionsType](#).) Which input properties are required and which tags are returned in the realm of shipping depends on whether the seller selected flat rate or calculated shipping, and on whether domestic or international shipping services have been offered.

With the Global Shipping Program, the international shipping provider determines the shipping parameters for the international leg.

Specifying Flat Rate Shipping

Set [ShippingDetails.ShippingType](#) to [Flat](#).

Specify the domestic and international shipping services. If any international shipping services are specified, at least one domestic shipping service must be specified. The seller can specify up to four domestic shipping services and up to five international shipping services from which the buyer may choose. Use [ShippingServicePriority](#) to note the seller's preference for sequence of display of the corresponding shipping service relative to the other services (1 = first in list).

Set the base shipping cost and the cost for shipping additional items, if fitting.

Set the insurance costs, if any. The values that the seller selects for such things as [ShippingServiceCost](#) and [InsuranceFee](#) are completely up to the seller, regardless of what the seller actually has to pay for such services. The seller sets the prices and the buyer chooses from among the shipping options. If the seller has offered insurance (as opposed to requiring it or not offering it at all), [InsuranceWanted](#) represents whether the buyer has chosen to pay for insurance. See "[Shipping Insurance](#)" on page 362.

If shipping insurance is used and [InsuranceFee](#) is a nonzero value, [InsuranceOption](#) should be one of [Optional](#), [Required](#) or [IncludedInShippingHandling](#). [IncludedInShippingHandling](#) exists because a carrier might include insurance at no additional cost when the shipping/handling cost reaches a certain level – see "[Automatic Shipping Insurance](#)" on page 365.

To learn how total shipping cost is calculated using these values, see "[UPS Rate Scales](#)" on page 349.

For details about the individual fields, see the eBay Unified Schema [eBay Trading API Call Reference](#) (for example, in [AddItem](#)):

[eBay Trading API Call Reference](#)

Maximum Flat Rate Shipping Cost

"Total shipping cost" is the sum of the seller's base flat rate shipping charge and the cost of insurance (if insurance is required). For some categories, eBay imposes a maximum on the shipping cost that a seller can specify for the first domestic flat rate shipping service.

In most cases, the maximum shipping cost depends only upon the category in which the listing is being created. The key element in such a case is [MaxFlatShippingCost](#).

In a few cases, the maximum cost depends not only on the category but also on which "shipping service group" the particular shipping service is in. eBay has put certain shipping services in groups. (Grouping is irrelevant to you and to the seller. But an example for a group established by eBay might be "all services for which insurance is required.") Key elements in these cases are three [Group](#)/[MaxFlatShippingCost](#) elements (where *N* can be 1, 2 or 3).

Call [GetCategoryFeatures](#) periodically (but not with every listing) to establish a database of facts about maximum flat rate shipping cost per site and per category and per shipping service. (See

[GetCategoryFeatures](#) documentation to learn the basic use of this call.) Here is how to interpret the response of [GetCategoryFeatures](#) regarding maximum flat rate shipping cost:

- If you find no [MaxFlatRateShippingCost](#) element and no [Group/MaxFlatShippingCost](#) elements, there is no maximum shipping cost for that category.
- If you find a [MaxFlatRateShippingCost](#) element, there is a maximum flat rate shipping cost for that category and it does not matter what shipping service was offered by the seller. (You will not find any [Group/MaxFlatShippingCost](#) elements if there is a [MaxFlatRateShippingCost](#) element.) The maximum cost is the value of the [MaxFlatRateShippingCost](#) element.
- If you find [Group/MaxFlatShippingCost](#) elements, the maximum flat rate shipping cost depends on what shipping group the first flat rate domestic shipping service is in. (You will not find any [MaxFlatRateShippingCost](#) element if there are [Group/MaxFlatShippingCost](#) elements.)

The value of each [Group/MaxFlatShippingCost](#) element is the maximum cost associated with the group with that name.

Call [GeteBayDetails](#) with [DetailName](#) set to [ShippingServiceDetails](#) and find the [ShippingServiceDetails](#) container corresponding to the first flat domestic shipping service. The value of [ShippingServiceDetails.CostGroupFlat](#) is the name of the group to which that shipping service belongs. The maximum cost is the value of the corresponding [Group/MaxFlatShippingCost](#) element from [GetCategoryFeatures](#).

- If [SiteDefaults.MaxFlatShippingCostCBTExempt](#) is returned and is [true](#), no maximum cost is imposed when the seller lists an item that ships into the country of that site from another country.

Once you know the maximum, you can test your seller's proposed total shipping cost to ensure it does not exceed the maximum shipping cost.

Using Shipping Rate Tables

On the eBay DE, UK and US websites, sellers can configure flat shipping rate tables in My eBay Shipping Preferences. These tables can then be applied to individual listings using the Trading API. Shipping rate tables enable sellers to tailor the flat shipping rates offered for an item to fit the shipping destination. They can specify a base rate for a country or an international region, then specify alternative rates or surcharges for shipping to more distant or inconvenient areas within the country or individual countries within the region.

Note: You can use the API to apply a shipping rate table to a listing, but to set up, modify or retrieve the contents of the table, sellers must use the ebay website. Currently, sellers can configure and apply one domestic rate table and one international rate table. For more information about configuring rate tables, see the [Specifying your shipping costs and locations help page](#).

Domestic Rate Tables

In a domestic shipping rate table, the seller can specify an alternative shipping rate for each shipping service category, in each of several domestic regions. For example, in the UK the shipping service categories are Economy, Standard and Express. In the US, they are Economy, Standard, Expedited and One-day.

Sellers use a dropdown list to specify the alternative shipping rates applied by the rate table as one of the following:

- A flat amount per item
- A flat surcharge
- A surcharge by weight

This selection applies to the entire table; there is no mixing and matching by region or shipping category.

The first row of the table's collapsible Domestic Regions section is the *base* region, which by default includes all domestic locations. The additional rows identify each of the more distant or inconvenient (*extended*) regions of the country to which the seller might also wish to ship. For example, in the UK these extended regions include Northern Ireland, the Scottish Highlands, and several different islands and island groups. In the US, the extended regions are Alaska and Hawaii, US Protectorates, and Army/Fleet Post Offices.

International Rate Tables

International rate tables work in much the same way as domestic rate tables, although an international rate table is available only on the eBay DE and UK sites. In an international rate table, the seller can specify an alternative shipping rate for each shipping service category, in each of several international regions. The shipping service categories are Economy, Standard and Express.

Sellers use a dropdown list to specify the alternative shipping rates applied by the rate table as one of the following:

- A flat amount per item
- A surcharge by weight

This selection applies to the entire table; there is no mixing and matching by region or shipping category.

The international rate table has nine top-level *base* regions: Africa, Asia, Central America and Caribbean, Europe, Middle East, North America, Oceania, Southeast Asia, and South America. Unlike in the domestic rate table, each base region is entirely comprised of its *extended* regions (countries). For example, the Southeast Asia base region consists of these extended regions: Brunei Darussalam, Cambodia, Hong Kong, Indonesia, Laos, Macau, Malaysia, Philippines, Singapore, Taiwan, Thailand, and Vietnam.

Configuring Shipping Rate Tables

When a seller enters or modifies a rate in a given shipping service category for a base region, that entry is automatically copied to all of its extended regions in the same category. The seller can then modify any of the extended region values in that category.

Note: When any extended region rate is changed, the base region rate is replaced with a reminder that the rates in that category (table column) are now specific to each extended region. Upon saving the table, if a value had previously been entered for the base position, that value is restored.

The rates entered in a shipping rate table replace (if a flat amount) or increase (if a surcharge) the shipping costs that were specified for the respective categories in the listing to which the table is applied.

The alternative rates specified in the table for a given shipping service category take effect only if the seller has offered a shipping service in that category for the listing. If a buyer is located in a region for which an alternative shipping rate was entered, that rate will be shown to the buyer.

Rates specified in the shipping rate table for a given region take effect only if the seller has not excluded that region, and only to the extent that the region has a rate specified in the table for a shipping service category that is offered in the listing. If the seller configures and applies a rate table that has no rate entered in any offered shipping service category for a particular region, and does not exclude the region, buyers in that region will be instructed to contact the seller for shipping costs and services.

Managing Rate Tables with the API

For listings made using the Trading API, use the `Item.ShippingDetails.RateTableDetails` container in the Add/Relist/Revise family of API calls to specify the `DomesticRateTable` or `InternationalRateTable` to use. Currently, sellers can define only one domestic and one international rate table, and the only applicable value of `DomesticRateTable` or `InternationalRateTable` is `Default`, which indicates that the currently defined table will be used. Both the domestic and international tables can be applied at the same time.

To get a list of the available domestic and international shipping locations, call `GeteBayDetails` with `DetailName` set to `ShippingLocationDetails`.

To specify the use of shipping rate tables with an item listing:

- 1 Call `GetUser` to ensure that the seller has configured the desired rate table on the eBay website (`User.SellerInfo.DomesticRateTable = true` or `User.SellerInfo.InternationalRateTable = true`).
- 2 Using the Add/Relist/Revise family of calls, for `Item.ShippingDetails.RateTableDetails`, specify the value `Default` for the `DomesticRateTable` or `InternationalRateTable` field to apply that rate table to the listing.

To stop using the rate table with the listing, provide an empty `<DomesticRateTable />` or `<InternationalRateTable />` element.

- 3 Specify flat rate shipping as you normally would, as described in “[Specifying Flat Rate Shipping](#)” on page 336.
- 4 If you are applying a shipping rate table that specifies a surcharge by weight, you must specify the item weight in the `ShippingDetails.CalculatedShippingRate` container's `WeightMajor` and `WeightMinor` fields, even though this is a flat rate listing. Do not use any other fields in the `ShippingDetails.CalculatedShippingRate` container because those are not supported in this scenario. Be aware that there is currently no way to determine through the API whether a seller's rate table specifies a surcharge by weight, so your application must make it clear to the seller that item weight must be supplied if the seller has specified a surcharge by weight.

Remember that any value in the `WeightMinor` field is rounded UP to the next unit, that is, to the next pound or kilogram, when the shipping cost is calculated, so make sure you message your users appropriately. One way to do this kind of UI messaging is the way the US eBay site does it,

by presenting sellers with a pulldown of weight ranges, where the pulldown lists, for example, 0 lb to 1 lb as one selection (1 lb), over 1 lb to 2 lbs as another selection (2 lbs), over 2 lbs to 3 lbs as another selection (3 lbs) and so on. However, your application can ask for and use specific weights (**WeightMajor** and **WeightMinor**) if you wish.

If the required weight values are not supplied in **ShippingDetails.CalculatedShippingRate**, a default weight of one unit (1 lb or 1 kg, depending on locale) is used as the basis for the surcharge. Currently the weight values used in this scenario are not returned by **GetItem** or **GetItemShipping**, although the values can be seen by the seller in the eBay UI.

For more information, see “[Package Dimensions and Weight](#)” on page 342.

- 5 Exclude any region to which the seller does not wish to ship even though that region has a service level and rate set up in the applied domestic or international shipping rate table. For example, a US seller might normally ship to Alaska and have Alaska shipping rates set up in the domestic rate table. However, if an item is too heavy and the shipping costs are too expensive, you can exclude Alaska for this item via the **ShippingDetails.ExcludeShipToLocation** field.

Specifying Calculated Shipping

Set **ShippingDetails.ShippingType** to **Calculated**.

Specify the domestic and international shipping services. If any international shipping services are specified, at least one domestic shipping service must be specified. The seller can specify up to four domestic shipping services and up to five international shipping services from which the buyer may choose. Use **ShippingServicePriority** to note the seller's preference for sequence of display of the corresponding shipping service relative to the other services (1 = first in list).

You should not attempt to set the base shipping cost (e.g. via **ShippingServiceCost**) because the base shipping cost is determined by eBay for the shipping service that the buyer selects from among the choices the seller has provided. (See “[Shipping Carriers and Services](#)” on page 314.)

If the seller wishes to add packaging/handling costs, set **CalculatedShippingRate.PackagingHandlingCosts**. See “[Packaging and Handling Cost](#)” on page 350.

You can offer or require insurance for an item with **InsuranceOption**. However, you cannot set **InsuranceOption** to **IncludedInShippingHandling** because insurance is not established by the seller but rather by eBay calculation for the selected shipping service.

The cost of insurance cannot be established for the shipping service until the sale price of the item is known. When it is established, it is seen in **ShippingInsuranceCost**.

See “[Shipping Insurance](#)” on page 362.

The sites which support domestic and international calculated shipping are United States, Canada, Canada France, and Australia.

Package dimensions and weight are required for some combinations of package type and shipping service on some sites. See “[Package Dimensions and Weight](#)” on page 342.

For details about the individual fields, see the eBay Trading API Call Reference (for example, in [AddItem](#)).

Package Dimensions and Weight

For calculated shipping, and for shipping based on a rate table that specifies weight-based surcharges.

Note: With the Global Shipping Program, the dimensions, weight and package type are required for calculated shipping on the domestic leg of the shipment; you can optionally provide these values for flat rate domestic shipping as well.

Measurement Units

The sites which support domestic and international calculated shipping are United States, Canada, Canada France, and Australia.

To specify the unit type of the weight and dimensions of a package, include `Item.ShippingDetails.CalculatedShippingRate.MeasurementUnit` in the call input. For best practice, use `MeasurementUnit`. Specifying `MeasurementUnit` and `measurementSystem` will cause the `measurementSystem` attribute to be ignored. The available options per site are:

- `US`: English (default)
- `CA`: Metric (default), English
- `CAFR`: Metric (default), English
- `AU`: Metric (default)

If no `MeasurementUnit` is specified, the default will be used.

Note: The Global Shipping Program is currently supported only on the eBay US site, so the unit type of measurement for Global Shipping packages (with either calculated or flat shipping) defaults to the United States value, `US`. As other sites make the Global Shipping Program available, you'll be able to set the measurement unit using `Item.ShippingPackageDetails.MeasurementUnit`.

Dimensions

The dimensions of a package are expressed with the following fields:

`CalculatedShippingRate.PackageDepth`
`CalculatedShippingRate.PackageLength`
`CalculatedShippingRate.PackageWidth`

These support the `unit` and `measurementSystem` attributes. The `unit` attribute is optional and is understood to be `in` (whole inches). The `measurementSystem` attribute is optional and is understood to be `English`. At this time, the only supported unit is whole inches.

The following would represent a package that is 3" x 8" x 8".

```
<PackageDepth>3</PackageDepth>  
<PackageLength>8</PackageLength>  
<PackageWidth>8</PackageWidth>
```

or, more formally:

```
<PackageDepth unit="in">3</PackageDepth>  
<PackageLength unit="in">8</PackageLength>  
<PackageWidth unit="in">8</PackageWidth>
```

or even:

```
<PackageDepth unit="in" measurementSystem="English">3</PackageDepth>  
<PackageLength unit="in" measurementSystem="English">8</PackageLength>  
<PackageWidth unit="in" measurementSystem="English">8</PackageWidth>
```

Developer impact: UPS requires dimensions on all UPS Air packages or Ground packages that are 3 cubic feet or larger, if they are to provide correct shipping cost. If package dimensions are not included for an item listed with calculated shipping, the shipping cost returned will be an estimate based on standard dimensions for the defined package type.

eBay enforces a dimensions requirement on listings so that buyers receive accurate calculated shipping costs.

Weights

The shipping weight of an item (or set of items shipped together) is expressed with **CalculatedShippingRate.WeightMajor** and **CalculatedShippingRate.WeightMinor**. These support the **unit** and **measurementSystem** attributes. The **unit** attribute is optional and is understood to be the **lbs** (pounds) for **WeightMajor** and **oz** (ounces) for **WeightMinor**. The **measurementSystem** attribute is optional and is understood to be **English**.

The following would represent a package weight of 5 lbs 2 oz:

```
<WeightMajor>5</WeightMajor>  
<WeightMinor>2</WeightMinor>
```

or, more formally:

```
<WeightMajor unit="lbs">5</WeightMajor>  
<WeightMinor unit="oz">2</WeightMinor>
```

or even:

```
<WeightMajor unit="lbs" measurementSystem="English">5</WeightMajor>  
<WeightMinor unit="oz" measurementSystem="English">2</WeightMinor>
```

Dimensions, Weight and Package Type for Global Shipping

Because the Global Shipping measurement system defaults to US, there's no need to include any measurement system or unit attributes with the dimensions or weight fields.

Global Shipping Package Dimensions

To specify the dimensions of a Global Shipping package, include values in whole inches for the following fields in the call input:

- `Item.ShippingPackageDetails.PackageDepth`
- `Item.ShippingPackageDetails.PackageLength`
- `Item.ShippingPackageDetails.PackageWidth`

Global Shipping Package Weight

To specify the shipping weight of a Global Shipping package, include values in the following fields:

- `Item.ShippingPackageDetails.WeightMajor`
- `Item.ShippingPackageDetails.WeightMinor`

For the eBay United States site, the unit value for `WeightMajor` defaults to lbs (whole pounds), and for `WeightMinor` it defaults to oz (whole ounces). For example, the following represents a package weight of 5 lbs 2 oz:

```
<WeightMajor>5</WeightMajor>
<WeightMinor>2</WeightMinor>
```

Global Shipping Package Type

To specify the type of package being shipped via the Global Shipping, include values for the following fields in the call input:

- `Item.ShippingPackageDetails.ShippingIrregular`
- `Item.ShippingPackageDetails.ShippingPackage`

`ShippingIrregular` is a boolean type. Set its value to `1` to indicate that the package is irregular and therefore cannot go through the stamping machine at the shipping service office, and thus requires special or fragile handling. Otherwise, set its value to `0`.

`ShippingPackage` is an enumerated type. Specify a value that appropriately describes the nature of the package.

When Package Dimensions and Weight are Required

For calculated shipping, package dimensions are required for irregular large packages or for packages of type `LargePackage` or if the shipping service is a UPS air service.

Dimensions or weight (or both) are required for certain combinations of package type and shipping service. You will need to evaluate certain data from `GeteBayDetails` to determine whether dimensions are required for a listing. On those eBay sites that support calculated shipping (US, CA, AU, eBay Motors Parts and Accessories), this applies only to domestic shipping. For the API, this

applies to domestic and international shipping for the US, CA and AU sites and eBay Motors Parts and Accessories.

Note: With the Global Shipping Program, package dimensions and weight are always required for calculated domestic shipping.

About the Elements for Dimensions and Weight Requirement

[GeteBayDetails](#) response

[ShippingServiceDetails](#)

[ShippingServicePackageDetails](#) ([ShippingServicePackageDetailsType](#))

[Name](#) ([ShippingPackageCodeType](#))

[DimensionsRequired](#) ([boolean](#); false if not returned)

[WeightRequired](#) ([boolean](#); false if not returned)

[ShippingPackageDetails](#)

[DimensionsSupported](#) ([boolean](#); false if not returned)

Note the difference in the two "package" container names: [ShippingServicePackageDetails](#) has "Service" in its name while [ShippingPackageDetails](#) does not. [ShippingServicePackageDetails](#) is about packages in the context of a particular service. [ShippingPackageDetails](#) is about packages in general.

[ShippingServicePackageDetails.DimensionsRequired](#) is a characteristic of a package type *in the context* of a particular shipping service.

[DimensionsSupported](#) is a characteristic of a *type of package*. There is no connection between it and individual shipping services.

(Trivia: [DimensionsSupported](#) will be [true](#) for a package type on a particular site if dimensions are required for that package type for *any* service on that eBay site. Also, if [DimensionsSupported](#) is [false](#) for a particular package type on that eBay site, [ShippingServicePackageDetails.DimensionsRequired](#) will be [false](#) for that package type for *all* services that support that package on that eBay site.)

Using the Elements for Dimensions and Weight Requirement

If the eBay site supports calculated shipping, use [GeteBayDetails](#) to determine whether package dimensions or weight are required for shipping services associated with that site. Call [GeteBayDetails](#) with one [DetailName](#) set to [ShippingPackageDetails](#) and with another [DetailName](#) set to [ShippingServiceDetails](#).

If [ShippingServicePackageDetails.DimensionsRequired](#) is [true](#), a seller who selects that package and service combination for a listing must submit [PackageWidth](#), [PackageLength](#) and [PackageDepth](#) in the item definition for the listing.

If [DimensionsSupported](#) is [true](#), a seller is allowed to provide package dimensions and eBay will store the seller-supplied dimensions, even if dimensions are not required by whatever service is ultimately selected by the buyer.

It is a best practice to always provide dimensions for a package, even if a particular combination of package type and selected shipping service does not require dimensions. This enables eBay and its shipping affiliates to be most accurate in its shipping cost calculations.

If `ShippingServicePackageDetails.WeightRequired` is `true`, a seller who selects that package and service combination for a listing must specify `WeightMajor` and `WeightMinor` in the item definition for the listing.

Examples

Scenario 1: Let's say that `DimensionsSupported` is `true` for the `LargePackage` package type for a particular site. And let's say that, for shipping services named S1 and S2, `ShippingServicePackageDetails.DimensionsRequired` is `true` for `LargePackage`. In this situation, the seller must provide dimensions.

Scenario 2: Let's say that `DimensionsSupported` is `true` for the `Package` package type for a particular site. And let's say that, for the shipping service named S1, `ShippingServicePackageDetails.DimensionsRequired` is `false` for `Package`. Further, for the shipping service named S2, `ShippingServicePackageDetails.DimensionsRequired` is `true` for `Package`.

In this case, if the seller only offers service S1, the seller can offer the `Package` option for service S1 and is not required to provide dimensions. (If the seller chooses to provide dimensions, eBay stores those dimensions and uses them for shipping cost calculations if the buyer selects service S1.)

In order for the seller to offer shipping service S2 for the `Package` package type, the seller must provide package dimensions.

Therefore, if the seller offers both services S1 and S2 and selects the `Package` option, the seller must provide package dimensions (to satisfy service S2's requirement).

Scenario 3: Let's say that `DimensionsSupported` is `false` for the package type `Letter`. And let's say that `ShippingServicePackageDetails.DimensionsRequired` is `false` for the `Letter` package type for all shipping services that the seller offers. In this case, dimensions are not needed at all; it would be best for the seller to not provide dimensions.

Scenario 4: Let's say that `ShippingServicePackageDetails.WeightRequired` is `true` for one of the shipping services offered by the seller and `false` for the rest. In this case, the seller must provide weight details for the item to satisfy that one service's requirement for weight details.

Specifying Freight Shipping

Freight shipping may be required if the items to be shipped exceed a particular weight. Freight shipping solutions exist only for the US, UK, Australia, Canada and Canada (French) sites, and only for domestic shipping.

The **eBay US site** provides a choice for freight shipping: either choose FreightQuote.com as the shipping carrier, or choose "other shipping service" (for freight shipping services not affiliated with eBay).

Note: The only way to choose FreightQuote.com is via the eBay US website. There is no API equivalent. In contrast, you can use either the website or the API to specify a shipping carrier other than FreightQuote.com.

When FreightQuote.com is the shipping carrier, eBay sets **ShippingType** to **Freight** and sets **ShippingService** to **FreightShipping**.

Note: The eBay UK site offers “courier” shipping. “Courier” is the UK’s name for what the US calls “freight”. Like in the US, the eBay Australia site and the eBay Canada sites also use the term “freight.”

If the site is US and FreightQuote.com is not the shipping carrier, OR if the site is UK and Courier was selected, OR if the site is Australia or Canada and Freight was selected, eBay sets **ShippingType** to **FreightFlat**.

For the US site, eBay sets **ShippingService** to **Freight**. For the UK site, eBay sets **ShippingService** to **Courier**. For the Australia site, eBay sets **ShippingService** to **AU_Freight**. For the Canada sites, eBay sets **ShippingService** to **CA_Freight**.

To specify “other freight service” on the eBay US site via API, provide only the following in the **ShippingDetails.ShippingServiceOptions** container:

```
<ShippingDetails>
  <ShippingServiceOptions>
    <ShippingService>Freight</ShippingService>
  </ShippingServiceOptions>
  <ShippingType>FreightFlat</ShippingType>
</ShippingDetails>
```

The table below summarizes the facts noted above.

Table 17-3 Specifying Freight Shipping via website and API

Freight option	Web solution	API solution
US site: To choose FreightQuote.com as the freight shipping service...	<p>What seller does:</p> <ul style="list-style-type: none"> - selects dropdown option Freight (instead of Flat or Calculated or No Shipping) - selects the radio button “FreightQuote.com” <p>What eBay does:</p> <ul style="list-style-type: none"> - sets ShippingType to Freight - sets ShippingService to FreightShipping - sets shipping cost according to FreightQuote.com’s calculation 	(None)

Table 17-3 Specifying Freight Shipping via website and API

Freight option	Web solution	API solution
US site: To choose a freight shipping service other than FreightQuote.com...	<p>What seller does:</p> <ul style="list-style-type: none"> - selects dropdown option Freight (instead of Flat or Calculated or No Shipping) - selects the radio button "Other freight service" <p>What eBay does:</p> <ul style="list-style-type: none"> - sets ShippingType to FreightFlat - sets ShippingService to Freight 	<p>What your code does:</p> <ul style="list-style-type: none"> - sets ShippingType to FreightFlat - submits an empty ShippingServiceOptions container - (optional and recommended) alters the item description to note that the seller and buyer will work out the shipping cost details outside of eBay <p>What eBay does:</p> <ul style="list-style-type: none"> - sets ShippingService to Freight - sets InsuranceDetails to NotOffered - sets shipping cost to 0; it is up to the seller and buyer to work out the cost
UK site: To specify freight shipping...	<p>What seller does:</p> <ul style="list-style-type: none"> - selects dropdown option Courier (instead of Flat or No Shipping) <p>What eBay does:</p> <ul style="list-style-type: none"> - sets ShippingType to FreightFlat - sets ShippingService to Courier 	<p>(The details here are the same as for the US above except for the name "Courier".)</p> <p>What your code does:</p> <ul style="list-style-type: none"> - sets ShippingType to FreightFlat - submits an empty ShippingServiceOptions container - (optional and recommended) alters the item description to note that the seller and buyer will work out the shipping cost details outside of eBay <p>What eBay does:</p> <ul style="list-style-type: none"> - sets ShippingService to Courier - sets InsuranceDetails to NotOffered - sets shipping cost to 0; it is up to the seller and buyer to work out the cost

Table 17-3 Specifying Freight Shipping via website and API

Freight option	Web solution	API solution
Australia site: To specify freight shipping...	<p>What seller does:</p> <ul style="list-style-type: none"> - selects dropdown option Freight (instead of Flat or No Shipping) <p>What eBay does:</p> <ul style="list-style-type: none"> - sets ShippingType to FreightFlat - sets ShippingService to AU_Freight 	<p>(The details here are the same as for the US above except for the name "AU_Freight".)</p> <p>What your code does:</p> <ul style="list-style-type: none"> - sets ShippingType to FreightFlat - submits an empty ShippingServiceOptions container - (optional and recommended) alters the item description to note that the seller and buyer will work out the shipping cost details outside of eBay <p>What eBay does:</p> <ul style="list-style-type: none"> - sets ShippingService to AU_Freight - sets InsuranceDetails to NotOffered - sets shipping cost to 0; it is up to the seller and buyer to work out the cost
Canada (English and French) site: To specify freight shipping...	<p>What seller does:</p> <ul style="list-style-type: none"> - selects dropdown option Freight (instead of Flat or No Shipping) <p>What eBay does:</p> <ul style="list-style-type: none"> - sets ShippingType to FreightFlat - sets ShippingService to CA_Freight 	<p>(The details here are the same as for the US above except for the name "CA_Freight".)</p> <p>What your code does:</p> <ul style="list-style-type: none"> - sets ShippingType to FreightFlat - submits an empty ShippingServiceOptions container - (optional and recommended) alters the item description to note that the seller and buyer will work out the shipping cost details outside of eBay <p>What eBay does:</p> <ul style="list-style-type: none"> - sets ShippingService to CA_Freight - sets InsuranceDetails to NotOffered - sets shipping cost to 0; it is up to the seller and buyer to work out the cost

UPS Rate Scales

Two rate scales are available for shipping when shipping is done via UPS:

- On-Demand Rates: standard UPS shipping rates.
- Daily Pick-Up Rates: reduced rates for high-volume shippers. Daily Pick-Up Rates translate into reduced shipping costs for the buyer when the seller elects to use calculated shipping.

Sellers identify which type of UPS rates they use by setting a preference on the Payment and Shipping Preferences web page. The selected rate type is indicated on the eBay website in the Shipping Calculator. It is seen in the API in **ShippingRateType**.

ShippingRateType is only returned if the value in **ShippingServiceOptions.ShippingService** is a UPS shipping service. If **ShippingRateType** has a value of **DailyPickup** on input and no UPS shipping service is used, there is no shipping costs reduction.

Search Results and Shipping Costs

The pages that eBay displays upon search can include details about shipping cost.

The shipping cost text displayed as a result of search will be "Not specified" if *all* of the following are true:

- The seller did not specify an **Item**-level **ShipToLocation** value of **None** (i.e. pickup only).
- The seller did not specify shipping type (i.e. **ShippingType** is not flat, calculated or freight shipping).
- The seller set **BuyerResponsibleForShipping** (only applicable for Motors vehicles) to false (note that the default for this field, if not passed, is true).
- The seller did not set **ShippingTermsInDescription** to true.

If an item's shipping costs are specified in the item description, you should set **ShippingTermsInDescription** to true when creating or revising a listing so that users of search on the eBay website will see something to the effect of "shipping terms in description" in the section about shipping costs.

For eBay Motors vehicles (but not for Parts and Accessories), passing **ShippingTermsInDescription** with a value of true has the same effect as setting **BuyerResponsibleForShipping** to false: the Search results page displays the shipping cost as "See description."

Specifying Other Costs

["Packaging and Handling Cost" on page 350](#)
["Shipping Surcharge" on page 351](#)
["Taxes" on page 351](#)

Packaging and Handling Cost

Note: The information here about the domestic packaging and handling cost for calculated shipping from the eBay US site is applicable to the domestic leg of a Global Shipping shipment. Handling cost for the international leg is included in the shipping cost determined by the international shipping provider.

Setting Packaging and Handling Costs for Calculated Shipping

Sellers assign a single packaging/handling costs for use by both domestic and international shipping services via **PackagingHandlingCosts**.

If domestic and international calculated shipping is offered for an item and if packaging/handling cost is specified only for domestic shipping, that cost will be applied by eBay as the international packaging/handling cost. (To specify a international packaging/handling cost, you must always specify a domestic packaging/handling cost, even if it is 0.)

International packaging/handling cost must be specified if calculated international shipping has been provided along with flat rate domestic shipping.

See also “[Packaging/Handling Cost for Combined Payment](#)” on page 361.

Revising an item

If a seller originally offered both domestic and international shipping services and assigned unique domestic and international packaging and handling costs, and now the seller revises the listing and omits the two packaging/handling costs, these costs will be zeroed. If the seller revises the listing and omits only the domestic packaging/handling costs, the domestic packaging/handling cost is zeroed. However, if the seller revises the listing and omits only the international packaging/handling costs, the value of the domestic packaging/handling cost is copied as the international packaging/handling cost. (If the seller for some reason wants to charge packaging/handling costs for the domestic shipping services but no packaging/handling costs for international packaging/handling costs, the seller should set the international to 0.0.)

If there are no shipping services, any packaging/handling costs are ignored if submitted.

Retrieving Packaging and Handling Costs for Calculated Shipping

GetItem: A seller can distinguish between domestic and international packaging/handling costs. The equation above, then, would be for a domestic shipping service. The equation for an international shipping service would use [InternationalPackagingHandlingCosts](#) instead of [PackagingHandlingCosts](#).

GetItemTransactions: Although [GetItem](#) returns both [PackagingHandlingCosts](#) and [InternationalPackagingHandlingCosts](#) for the eBay US and Motors and Parts and Accessories sites, there is no need for order retrieval calls such as [GetItemTransactions](#), [GetSellerTransactions](#), and [GetOrderTransactions](#) to return separate tags. Order retrieval calls reflect the selection of the buyer, whether the selection was of a domestic shipping service and its costs or an international shipping service and its costs. Packaging/handling costs corresponding to the buyer's selection are adequately reflected in the [PackagingHandlingCosts](#) field.

GetItemShipping: Any packaging/handling cost specified on input is added to each shipping service on output (whether domestic or international).

Shipping Surcharge

Sellers can specify a surcharge when shipping via UPS to Alaska, Hawaii or Puerto Rico. This is currently available only for flat rate shipping on the eBay US site and Motors Parts and Accessories categories.

Note: Shipping surcharges do not apply to shipments via the Global Shipping Program.

Taxes

Total item cost to the buyer includes such things as item cost, shipping cost, taxes, and discounts. The application developer must determine if taxes apply and whether they are based on the item cost without shipping costs, on the item cost with shipping costs, etc. See “[Enabling Multi-jurisdiction Sales Tax](#)” on page 231.

Determining Shipping Costs for a Listing

The responses of several calls such as [GetItem](#), [GetItemTransactions](#), [GetItemShipping](#), and [GetOrder](#) include shipping information.

["Key Calls for Determining Shipping Costs" on page 352](#)

["Determining Flat Rate Shipping Costs" on page 353](#)

["Determining Calculated Shipping Costs" on page 354](#)

["Determining Freight Shipping Cost" on page 355](#)

Key Calls for Determining Shipping Costs

Various Get calls, such as [GetItem](#) and [GetItemTransactions](#), return [ShippingDetails.ShippingType](#).

GetItem

Use this call to retrieve shipping costs for an item at any point in the life of the listing. Note, however, that if you use it before the listing ends, information based on final item price, such as cost of insurance, cannot be known and will not be returned.

GetOrders

Use this call to retrieve actual shipping costs. Final item price is known and information based on final item price, such as cost of insurance, can be established by eBay and returned, and you can determine total cost.

GetItemShipping

This call is analogous to the Shipping Calculator seen in both the buyer and seller web pages. It is another way for you to obtain shipping cost information for an item at any point in the life of a listing for multiple shipping services. It is only for items for which calculated shipping is offered. Thus, insurance costs can only be known if the item has sold and item final price is known.

GetItemShipping with Global Shipping

International buyers can use [GetItemShipping](#) to retrieve Global Shipping Program information for a listing. Provide the [DestinationCountryCode](#) and the [DestinationPostalCode](#) in the input to get the following relevant output fields:

Example 17-2 GetItemShipping Output for Global Shipping

```
<InternationalShippingServiceOption>
  <ShippingService>InternationalPriorityShipping</ShippingService>
  <ShippingServiceCost currencyID="USD">40</ShippingServiceCost>
  <ImportCharge currencyID="USD">40</ImportCharge>
  <ShipToLocation>GBR</ShipToLocation>
  <ShipToLocation>CA</ShipToLocation>
</InternationalShippingServiceOption>
```

The [ShippingServiceCost](#) and [ImportCharge](#) fields comprise the total amount the buyer must pay to ship the item. [ImportCharge](#) is not returned unless the [ShippingService](#) field has the value [InternationalPriorityShipping](#).

GetShippingCosts

This is a Shopping API call similar to [GetItemShipping](#). It provides the same `ShippingDetails.InternationalShippingServiceOption` container as [GetItemShipping](#), but also includes the `ShippingCostSummary` container, which provides the same `ShippingServiceCost` and `ImportCharge` fields, along with fields for `ListedShippingServiceCost` and `ShippingType`. For more information, see the [Shopping API Call Reference](#).

GetShippingCosts with Global Shipping

International buyers can use [GetShippingCosts](#) to retrieve Global Shipping Program information for a listing. Provide the `DestinationCountryCode` and the `DestinationPostalCode` in the input to get the following relevant output fields:

Example 17-3 GetShippingCosts Output for Global Shipping

```
<InternationalShippingServiceOption>
  <ShippingServiceName>InternationalPriorityShipping</ShippingServiceName>
  <ShippingServiceCost currencyID="USD">40</ShippingServiceCost>
  <ShippingServicePriority>0</ShippingServicePriority>
  <ShipsTo>GBR</ShipsTo>
  <ShipsTo>CA</ShipsTo>
  <ImportCharge currencyID="USD">40</ImportCharge>
</InternationalShippingServiceOption>
...
<ShippingCostSummary>
  <ShippingServiceName>InternationalPriorityShipping</ShippingServiceName>
  <ShippingServiceCost currencyID="USD">40</ShippingServiceCost>
  <ShippingType>Calculated</ShippingType>
  <ListedShippingServiceCost currencyID="USD">40</ListedShippingServiceCost>
  <ImportCharge currencyID="USD">40</ImportCharge>
</ShippingCostSummary>
```

The `ShippingServiceCost` and `ImportCharge` fields comprise the total amount the buyer must pay to ship the item. `ImportCharge` is not returned unless the `ShippingServiceName` field has the value `InternationalPriorityShipping`. The `ShipsTo` field is the Shopping API equivalent of `ShipToLocation`.

Determining Flat Rate Shipping Costs

Note: The information in this section about domestic flat shipping is applicable to the domestic leg of a Global Shipping shipment. Shipping costs for the international leg are determined by the international shipping provider.

The total shipping cost (assuming the buyer chooses to pay for shipping insurance) is

$$\text{ShippingServiceCost} + (\text{ShippingServiceAdditionalCost} * (\text{QuantitySold} - 1)) + \text{InsuranceFee} + \text{ShippingSurcharge}$$

(where you use the domestic version of tags if the buyer selects a domestic shipping service and the international version of tags if the buyer selects an international shipping service).

For example, assume a multi-item fixed price listing with 10 items is listed with a **ShippingServiceCost** of \$8.00, a **ShippingServiceAdditionalCost** of \$2.00, and no insurance charge or surcharge. If a buyer buys four items from the listing, the total shipping cost is:

$$\$8 + (\$2 * 3) + \$0 + \$0 = \$14$$

Note: The order line item-based calls ([GetItemTransactions](#), [GetSellerTransactions](#), and [GetOrderTransactions](#)) reflect the selection of the buyer, whether the selection was of a domestic shipping service and its costs or an international shipping service and its costs. Thus, insurance details corresponding to the buyer's selection are reflected in the (non-container) tags **InsuranceFee** and **InsuranceOption**.

For details about the individual fields, see the eBay Unified Schema [eBay Trading API Call Reference](#) (for example, in [GetItem](#)):

[eBay Trading API Call Reference](#)

Address Change

If an application enables the buyer to change the shipping address outside of eBay, the application should call eBay to determine the shipping cost, since **ShippingSurcharge** may apply. See "[Shipping Surcharge](#)" on page 351.

Determining Calculated Shipping Costs

Note: The information in this section about domestic calculated shipping is applicable to the domestic leg of a Global Shipping shipment. Shipping costs for the international leg are determined by the international shipping provider.

The total shipping cost (assuming the buyer chooses to pay insurance) is

$$((\text{ShippingServiceCost} + \text{ShippingInsuranceCost}) * \text{QuantitySold}) + \text{PackagingHandlingCosts}$$

(where you use the domestic version of tags if the buyer selects a domestic shipping service and the international version of tags if the buyer selects an international shipping service).

ShippingServiceCost is the cost to ship without insurance, taxes, discounts, etc. It is a value calculated for a specific shipping service. **ShippingServiceAdditionalCost** is always 0.00 for calculated shipping.

See "[Retrieving Insurance Costs for Calculated Shipping](#)" on page 364 and "[Retrieving Packaging and Handling Costs for Calculated Shipping](#)" on page 351.

For [GetItemTransactions](#), the quantity of items is already factored into the values of **ShippingServiceCost** and **ShippingInsuranceCost**. (Effectively, **QuantitySold** should have a value of 1 in the calculation above.)

[GetItemTransactions](#) also returns the following, since item final price and shipping service are known: **InsuranceWanted**, the **ShippingServiceSelected** container, **ShippingInsuranceCost** (domestic and international), **ShippingRateErrorMessage**, **ShippingRateType**, **ShippingServiceUsed**.

For details about the individual fields, see the eBay Unified Schema [eBay Trading API Call Reference](#) (for example, in GetItem):

[eBay Trading API Call Reference](#)

Determining Freight Shipping Cost

Freight shipping cost is found in `ShippingDetails.ShippingServiceOptions.ShippingServiceCost`.

It is non-zero for a listing only if the item is on the eBay US site and FreightQuote.com was selected as the freight shipping service.

Other Shipping Features

[“Ensuring PayPal Seller Protection” on page 355](#)

[“Shipping Cost Discount Profiles” on page 356](#)

[“Specifying the Cash on Delivery Option in Shipping Details” on page 361](#)

[“Shipping Insurance” on page 362](#)

Ensuring PayPal Seller Protection

To ensure that a seller is covered by PayPal Seller Protection, the seller should ship to the address specified on the Transaction Details page of their PayPal account.

See also:

[PayPal seller protection for eBay](#)

<http://pages.ebay.com/paypal/seller/protection.html>

[Additional PayPal seller protection information](#)

<https://www.paypal.com/cgi-bin/webscr?cmd=xpt/Marketing/securitycenter/sell/SellerProtection-outside>

Shipping Cost Discount Profiles

In this section:

- “Introduction” on page 356
- “Discount Rules and Discount Profiles” on page 357
- “Profile Areas” on page 357
- “Variable Rules and Fixed Rules” on page 357
- “Creating Shipping Discount Profiles” on page 358
- “Shipping Insurance Requirement for Combined Payment” on page 358
- “Packaging/Handling Discount Profile Requirement for Calculated Shipping” on page 358
- “Assigning Shipping Discount Profiles” on page 359
- “Miscellaneous Related Topics” on page 359
- “Modifying Shipping Discount Profiles” on page 360
- “Deleting Shipping Discount Profiles” on page 360
- “Key API Mechanisms” on page 361
- “Shipping Insurance for Combined Payment” on page 361
- “Packaging/Handling Cost for Combined Payment” on page 361

Introduction

eBay enables a seller to offer various kinds of shipping cost discounts on orders (items purchased separately from the same seller and now combined for payment and shipping).

Note: The shipping costs of orders shipped through the Global Shipping Program are consolidated into a single value consisting partly of domestic shipping and handling charges, and partly of international shipping and handling charges that are controlled by the international shipping provider. Sellers can discount only the domestic portion of the total shipment.

The kind of discount that an item can receive in that order is determined by what shipping cost discount profile was assigned to the item when it was listed, and, if the seller defined a global promotional discount and it was in effect at the time the order line item was created, the order can also benefit from a promotional discount. The discounts “applied to the item” are only realized when the item is combined in an order—no shipping discount is applied to the item if it is purchased and shipped by itself.

A user can also offer a packaging/handling cost discount and can specify shipping insurance costs for combined payment.

Discount Rules and Discount Profiles

In the API, *discount rules* are the kinds of discounts that are made available to sellers. The following are loose descriptions of the discount rules that are available for flat rate shipping.

- “Charge the highest shipping cost, and add the specified amount for each additional item.”
- “Charge the highest shipping cost, and each additional item ships for free.”
- “Charge the highest shipping cost, and subtract the specified amount for each additional item.”
- “Charge the highest shipping cost, and subtract the specified percentage for each additional item.”

These descriptions reflect amounts and percentages. If a user wishes to signify that an item qualifies for a shipping discount, the user must first create a shipping discount *profile*. That profile will be based on one of the shipping discount *rules* made available by eBay. Rules are provided by eBay; profiles are created by users and are based on such rules.

The profile definition would need to include any amount or percentage that corresponds to the rule. For example, the API rule corresponding to the first description above is named `EachAdditionalAmount`. A user can create a flat rate shipping discount profile (and make up a name for the profile, such as “flat ship 2”) whose `DiscountName` is `EachAdditionalAmount`, and the user specifies the fee in `DiscountProfile.EachAdditionalAmount`. A user can create additional profiles for this same rule, each with a unique fee.

Profile Areas

These are the major areas for which profiles can be created: calculated shipping discounts, flat rate shipping discounts, packaging/handling cost discounts, and promotional discount.

A user can define, at most, one promotional shipping cost discount.

A user can define, at most, one packaging/handling cost discount.

A user can have zero or more profiles for each of the other profile areas (if the profile is based on a variable rule, as discussed in “[Variable Rules and Fixed Rules](#)” on page 357).

In the profile areas for which a user can have multiple discount profiles, all profiles must be based on the same underlying rule. For example, `FlatShippingDiscount` has several rules, one of which is `EachAdditionalAmountOff`. The user can have multiple profiles based on that rule. If the user wishes to create flat rate shipping discount profiles based on the `EachAdditionalPercentOff` discount rule, the user would first need to delete the other profiles for flat rate shipping discounts.

Variable Rules and Fixed Rules

A variable rule is a rule which can be used in a discount profile only if accompanied by a value for amount or percentage. For example, for a user to create a calculated shipping discount profile based on the `WeightOff` rule, a variable rule, the user must specify the amount of weight in the `WeightOff` field.

In contrast, no additional value is needed when the profile is based on a fixed rule. For example, if the user creates a profile based on the fixed rule named `CombinedItemWeight` and then assigns that

profile to several listings, the user is designating that the calculated shipping cost for combined payment would be based on the combined weights of all items.

This distinction is important because a user can have one or more profiles based on the same variable rule, or the user can have one profile based on a fixed rule, but the user cannot have both. If the user has several profiles based on a variable rule and now the user wishes to switch to a fixed rule for subsequent item listings, the user must first delete all profiles based on the variable rule before the user can create the profile based on the fixed rule.

Creating Shipping Discount Profiles

A user can define shipping discount profiles via the eBay website (My eBay, Combined Payment) and via [SetShippingDiscountProfiles](#). (Once a user has defined profiles, the user can assign one or more to listings. See “[Assigning Shipping Discount Profiles](#)” on page 359.)

When creating profiles via [SetShippingDiscountProfiles](#), the user always identifies the underlying discount rule by name. If it is a variable rule, the user can create as many profiles at the same time via [SetShippingDiscountProfiles](#) as he wishes, provided they are all based on the same rule. The user must set the value of whatever field corresponds to the choice of variable rule. For example, if the user selected [WeightOff](#) as the rule for a new calculated shipping cost discount profile, the user would also need to set the number of ounces in [DiscountProfile.WeightOff](#).

The [SetShippingDiscountProfiles](#) has a [DiscountProfile.DiscountProfileID](#) field. This field is not used when a user creates a profile but only when modifying or deleting a profile. eBay assigns a unique ID to every new profile. You can get the IDs for a user’s profiles by calling [GetShippingDiscountProfiles](#).

The user can create, at most, one profile based on a fixed rule (provided the user has first deleted all profiles for that profile area that are based on a variable rule).

If this is the first and only profile the user is creating for a particular profile area in the call to [SetShippingDiscountProfiles](#), the user cannot assign a name to the profile.

At least one profile must exist for that profile area for the user to be able to assign unique names to subsequent profiles in that profile area. The name that the user assigns to a profile is completely up to the user.

To “rename” the first “nameless” profile after additional profiles have been created, the user must create a new profile whose characteristics match those of the first profile plus include a name, and then the user must delete the original profile.

Shipping Insurance Requirement for Combined Payment

Shipping insurance cost must be specified for combined payment if the user wishes to define either flat rate shipping cost discount profiles or a promotional shipping discount profiles. Either the user must define both domestic and international shipping insurance costs before creating any of those profiles, or the user must define those insurance costs in the same call of [SetShippingDiscountProfiles](#) that creates the new profiles. (This requirement does not apply for eBay sites for which shipping insurance does not apply.)

Packaging/Handling Discount Profile Requirement for Calculated Shipping

A packaging/handling profile must be created if a calculated shipping cost discount profile is to be created. Either the user must define it before creating any calculated shipping cost discount profile,

or the user must define it at the same time the first calculated shipping cost discount profile is created.

Assigning Shipping Discount Profiles

Once shipping cost discount profiles have been defined, a seller can assign them to listings, relistings and revised listings. If a listing offers flat rate shipping services, the seller can assign, at most, one flat rate shipping cost discount profile. Similarly, a seller can assign one calculated shipping cost discount profile to a listing that offers calculate shipping. Regardless of whether any flat rate or calculated shipping cost discount profile has been assigned to the listing, the seller can apply the promotional shipping cost discount, if such has been defined. The seller can also assign a packaging/handling cost discount profile.

In order to assign a shipping cost discount profile, the seller must have the ID of the profile. You can get the names and IDs of available shipping cost discount profiles with [GetShippingDiscountProfiles](#), a call that returns details about the shipping cost discount profiles defined by the user.

When listing or relisting or revising a listing, set the appropriate [ShippingDetails](#) members:

- **ShippingDiscountProfileID** – the ID of the shipping discount to offer for the domestic shipping services (where the shipping discount is either of type FlatShippingDiscount or CalculatedShippingDiscount)
- **InternationalShippingDiscountProfileID** – the ID of the shipping discount to offer for the international shipping services (where the shipping discount is either of type FlatShippingDiscount or CalculatedShippingDiscount)
- **PromotionalShippingDiscount** – a boolean signifying that the global promotional discount is to apply to domestic shipping for this listing if it is combined into an order

In order for a promotional discount to be applied to an item in an order, the promotional discount must have been in effect when the items was listed, and the item must have been opted into that promotional discount, and the promotional discount must have been in effect when the order line item was created.

- **InternationalPromotionalShippingDiscount** – a boolean signifying that the global promotional discount is to apply to international shipping for this listing if it is combined into an order

If the seller sets **ShippingDiscountProfileID** to a calculated shipping cost discount profile, the current profile for packaging/handling cost discount is applied. (There is therefore no field in [AddItem](#) for identifying the packaging/handling cost discount profile.)

Miscellaneous Related Topics

Shipping discount profiles can be assigned to multi-quantity listings. The buyer receives whatever works out to be the best combination of values. For example, if the non-discounted packaging/handling fee for individual items would be \$5 for the first item and \$4 for each additional item, and if a packaging/handling cost discount applies, one for which the cost is \$3 for each additional item, the better of these, \$3, is the fee for additional items, in the context of combined payment.

[GetItem](#) returns shipping cost discount details for those items and orders for which shipping cost discounts were applied.

The amount saved in an order for which shipping cost discounts apply is returned in the **AmountSaved** field of [GetOrders](#) and [GetOrderTransactions](#).

If an order qualifies for shipping discounts and was created via the eBay website, one more shipping service is presented to the buyer (in addition to whatever shipping services the seller is offering to the buyer for the order). This additional shipping service is named Promotional Shipping Service. This service is added by eBay when order details are retrieved for the buyer (it is not, for example, something that the seller can select somewhere). A buyer can select one or none of the services being offered. Thus, if the buyer selects Promotional Shipping Service, the buyer is selecting the discounted shipping cost and the choice of shipping carrier is up to the seller. Conversely, if the buyer selects one of the other shipping services, the buyer does not get the shipping discount.

Modifying Shipping Discount Profiles

To modify a shipping discount profile, call [SetShippingDiscountProfiles](#) with **ModifyActionCode** set to **Update** and provide all details for the updated profile. If part of the modification is to change from a variable rule to a fixed rule, you must first delete all profiles based on the variable rule. See “[Variable Rules and Fixed Rules](#)” on page 357.

Deleting Shipping Discount Profiles

To delete a flat rate or calculated shipping discount profile, first call [GetShippingDiscountProfiles](#) to obtain valid **DiscountProfileIDs**, and then call [SetShippingDiscountProfiles](#) with **ModifyActionCode** set to **Delete** and set **DiscountProfileID** to the **DiscountProfileID** of the profile to be deleted. When you delete a profile, the only listings that are affected are active listings for which no bids have been made and which were associated with the discount profile that is being deleted. Listings for which at least one bid has been made remain unaffected: they continue with the discount that was originally assigned to the listing. For those active listings for which no bids have been made, you can set **MappedDiscountProfileID** to the ID of the profile that is to be used in place of the profile being deleted.

Once a user has defined a packaging/handling profile, the user cannot delete it. However, the user can change the underlying rule in [CalculatedHandlingDiscount.DiscountName](#) to [CombinedHandlingFee](#) or [IndividualHandlingFee](#). Call [SetShippingDiscountProfiles](#) with **ModifyActionCode** set to **Update**.

To “delete” the shipping insurance for combined payment, set **InsuranceOption** to **NotOffered**. It is not necessary to redefine the range/cost pairs.

Promotional discounts may not be deleted through the API. To delete a promotional discount, remove it through the eBay web interface at Home > My eBay > My Account > Preferences > Combined Payments and Shipping Discounts.

Deleting a Shipping Discount Profile from an Item When Relisting or Revising

If a listed item has a Shipping Discount Profile defined, you can remove that discount profile in the Relist or Revise item family of calls. To do this, supply the entire shipping container (**ShippingDetails**) and for the **ShippingDiscountProfileID** or **InternationalShippingDiscountProfileID** supply the value 0 (zero). This will result in the removal of that shipping discount profile from that item.

Key API Mechanisms

[GetShippingDiscountProfiles](#)
[SetShippingDiscountProfiles](#)
[GetShippingDiscountProfiles.FlatShippingDiscount.DiscountProfile](#)
[GetShippingDiscountProfiles.CalculatedShippingDiscount.DiscountProfile](#)
[GetShippingDiscountProfiles.PromotionalShippingDiscount](#)
[GetShippingDiscountProfiles.PromotionalShippingDiscountDetails](#)
[SetShippingDiscountProfiles.PromotionalShippingDiscountDetails](#)

Shipping Insurance for Combined Payment

The [ShippingInsurance](#) and [InternationalShippingInsurance](#) in [SetShippingDiscountProfiles](#) (for domestic and international orders, respectively) enable the user to pair ranges of item price with shipping insurance fee. See:

[SetShippingDiscountProfiles.ShippingInsurance](#)
[SetShippingDiscountProfiles.InternationalShippingInsurance](#)

Packaging/Handling Cost for Combined Payment

The user can specify how packaging/handling cost is to be determined globally for orders via the [CalculatedHandlingDiscount](#) field of [SetShippingDiscountProfiles](#). See:

[SetShippingDiscountProfiles.CalculatedHandlingDiscount.DiscountName](#)

Specifying the Cash on Delivery Option in Shipping Details

On the Italy site (site ID 101), a seller can provide an improved Cash on Delivery (COD) option. Using [AddItem](#), a seller can enable a buyer to choose COD as the shipping method.

To list with COD for shipping requires the following:

- The listing is made on the Italy site (site ID 101).
- The payment methods include the COD payment method.
- A shipping service is included that supports COD and that is an Italy-domestic shipping service (e.g., [IT_ExpressPackageMaxi](#)). You can use [GeteBayDetails](#) to obtain an applicable shipping service.
- The cost of COD shipping is specified in the request (see below) and is greater than zero.

A seller who is using [AddItem](#) to enable the COD shipping option would include the following in the [AddItem](#) request:

Example 17-4 Including Cash on Delivery as a Shipping Option

```
...
<Item>
  ...
  <Country>IT</Country>
  <PaymentMethods>COD</PaymentMethods>
  <ShippingDetails>
    ...
    <ShippingServiceOptions>
      <ShippingServicePriority>1</ShippingServicePriority>
      <ShippingService>IT_ExpressPackageMaxi</ShippingService>
      <ShippingServiceCost>5</ShippingServiceCost>
      <ShippingServiceAdditionalCost>1</ShippingServiceAdditionalCost>
    </ShippingServiceOptions>
    <CODCost currencyID="EUR">2.0</CODCost>
  </ShippingDetails>
  ...
</Item>
...
```

A seller can specify COD as a shipping option in the following calls:

- [AddItem](#), [ReviselItem](#), [RelistItem](#), and [VerifyAddItem](#)
- [AddOrder](#)
- [ReviseCheckoutStatus](#)
- [SendInvoice](#)

The following calls can include COD as the shipping method for a listing:

- [GetItem](#)
- [GetItemShipping](#)
- [GetItemTransactions](#)
- [GetOrders](#)
- [GetOrderTransactions](#)
- [GetSellerTransactions](#)

Shipping Insurance

Note: With request version 635 and higher, the international and domestic shipping insurance option is available only on the following sites: AU, FR, and IT. If you attempt to list an item with a shipping insurance option on a site that does not support the shipping insurance option, the shipping insurance option is removed and a warning is issued.

Setting Insurance Costs for Flat Rate Shipping

Domestic insurance is represented by this container:

```
<InsuranceDetails>
  <InsuranceFee>
  <InsuranceOption>
</InsuranceDetails>
```

International insurance is represented by this container:

```
<InternationalInsuranceDetails>
  <InsuranceFee>
  <InsuranceOption>
</InternationalInsuranceDetails>
```

Note that if you include buyer-paid shipping insurance on an item (listed on one of the sites that supports the shipping insurance option), a buyer on a site that does not support buyer-paid shipping insurance can still purchase that item. In these cases, the buyer is responsible for all the shipping insurance terms that have been outlined in the item listing.

It's worth mentioning that eBay has found that buyer-paid shipping insurance causes negative experiences for both buyers and sellers alike. Buyers are confused as to why they need to purchase shipping insurance, especially in light of PayPal's Buyer Protection service (which applies whether or not a seller uses shipping insurance). Note that sellers are responsible for the items they sell until they safely arrive in their customers' hands. Offering buyer-paid insurance (either as an optional or required service) infers that the buyer is somehow responsible for the safe delivery of the items they purchase. This notion can reduce buyer confidence in the marketplace.

eBay understands that it is in the seller's best interest to insure certain shipments (for example, antiques, fragile items, and so on). If appropriate for the items they sell, and for the business they run, sellers can choose to purchase insurance for the items they ship. While completely optional, sellers can fold any insurance costs into the price of their items, either by adding the cost to the item price or by adding it as a handling charge. In these cases, sellers can include a statement in their item descriptions that clarify the shipping details (for example, "shipping insurance provided").

Requirements and restrictions:

- **InsuranceFee** can only be specified if there is at least one shipping service.
- **InsuranceFee** is the seller-defined cost of shipping insurance. If shipping insurance will not be used, simply omit the **InsuranceOption** and **InsuranceFee** tags.
- If **InsuranceOption** is set to **Required** or **Optional**, **InsuranceFee** must be specified.
- If both domestic and international shipping services are offered but insurance is only specified for the domestic services, **InsuranceOption** for international shipping will automatically be set to **NotOffered**.
- If no shipping services or shipping cost are provided, any insurance details provided are dropped.
- If no domestic shipping services are provided, any domestic insurance provided will be dropped. Similarly, if no international shipping services are provided, any international insurance provided will be dropped.

See also “[Shipping Insurance for Combined Payment](#)” on page 361.

Revising an item

If the item already has insurance and no additional insurance details are provided, the existing insurance values are not modified.

If the item already has insurance and additional insurance tags are passed in, only the corresponding values are updated.

If the seller revises the listing to remove domestic or international shipping services, **InsuranceOption** is set to **NotOffered** and **InsuranceFee** is not returned in the Get calls.

Retrieving Insurance Costs for Flat Rate Shipping

If shipping services were offered, **GetItem** returns the following tags and containers:

InsuranceDetails.InsuranceOption
InsuranceDetails.InsuranceFee
InternationalInsuranceDetails.InsuranceOption
InternationalInsuranceDetails.InsuranceFee

InsuranceDetails and its tags represents insurance for the domestic shipping services.

InternationalInsuranceDetails and its tags represents insurance for the international shipping services.

If domestic insurance was not specified for the item, **InsuranceDetails.InsuranceOption** is set to **NotOffered** and **InsuranceDetails.InsuranceFee** will not be returned. Similar is true for international insurance.

Retrieving Insurance Costs for Calculated Shipping

GetItemTransactions: The insurance cost returned for the shipping service is for a single item. Therefore, if the buyer purchased multiple items, you must multiply **ShippingInsuranceCost** by the number of items to arrive at the total shipping cost. See “[Shipping Insurance](#)” on page 362.

GetOrders: The insurance cost returned for the shipping service is for the entire order. See “[Shipping Insurance](#)” on page 362.

Automatic Shipping Insurance

There is a condition in which a buyer receives shipping insurance for free. If the buyer was anticipating that he could request shipping insurance for the item and then does not find this choice, this may confuse the buyer. This section describes that condition so that you can consider how to make this clear to a buyer when this condition arises.

If the seller selects calculated shipping when listing the item, the cost of shipping insurance is determined as part of the shipping cost calculations that occur when the item has been sold (the point when the final item value and the buyer's zip code are known).

Some shipping carriers (such as USPS and UPS) automatically include free shipping insurance for items with a value below a certain carrier-set amount, regardless of whether the seller specified that insurance is required, optional or not offered. eBay, working with the shipping carrier, automatically detects whether the item price is less than the carrier's threshold value and whether shipping insurance should automatically be made part of the order line item (at no additional charge). Since this might happen when a buyer is anticipating the option to request shipping insurance, this is a point of potential confusion for the buyer. Thus, it is worth your considering how to make this clear to the buyer when the situation arises.

Chapter 18

Specialty Categories and Listings

This chapter describes considerations that you should be aware of when you work with items in certain specialty categories, items with special characteristics, or items that eBay presents within tailored shopping experiences.

- “[Classified Ad Listings](#)” on page 366
- “[eBay Real Estate Listings](#)” on page 367
- “[Digital \(Downloadable File\) Listings](#)” on page 368
- “[Ticket Listings](#)” on page 370

Also see these chapters for information about working with specialty eBay sites:

- “[Managing eBay Stores](#)” on page 534
- “[eBay Motors](#)” on page 559

Classified Ad Listings

The Classified Ad (advertisement) listing format was created for sellers who want to get in contact with as many interested buyers as possible. As with regular auctions, when the seller specifies a start price, it only serves as an asking price. No bidding takes place.

Important: This section discusses how to use the Trading API to manage classified ad listings on [eBay.com](#) websites. It cannot be used to list anything on [eBayClassifieds.com](#), which is a different website. Classified ads on [eBay.com](#) are different from classified ads on [eBayClassifieds.com](#), which is not supported by the Trading API. For information about the other differences between these two sites, see [Advertising with classified ads on eBay and eBay Classifieds](#).

With the Classified Ad format, interested buyers fill in a contact form and their information is sent to the seller. Classified Ad format listings do not result in feedback, because the format aims to generate multiple leads rather than a single high bidder. You can use [GetCategoryFeatures](#) to determine whether support for your listing type is available in the category in which you want to list.

When the Ad format is used, no auction takes place. Instead, a user can leave a *lead* for an item to show interest. An application can retrieve a seller’s leads using [GetAdFormatLeads](#). To list a property in the Ad format, set the value of [Item.ListingType](#) to [LeadGeneration](#) and [Item.ListingSubtype2](#) to [ClassifiedAd](#) when invoking [AddItem](#). Some sites, such as the eBay Germany site, only support Ad format listings (no Real Estate auctions).

In upcoming months on the US site, the pay-per-lead feature will enable you to use pay-per-lead functionality in classified ads. If a given category supports pay-per-lead listings, they will be created using [Item.ListingDetails.PayPerLeadEnabled](#) (as well as fields in the [Item.ExtendedSellerContactDetails](#) container) in [AddItem](#).

The following differences are applicable for Classified Ad format listings:

- Once you have specified that the listing is a Classified Ad format listing, you cannot change the listing to another format when revising or relisting the item.
- eBay Classified Ad format listings can have durations of 30 or 90 days.
- eBay Germany Classified Ad format listings can have durations of 10 or 30 days.

The following **Item** properties are not applicable for Classified Ad format listings:

- All payment methods ([PaymentMethods](#))
- [Storefront](#)
- [ReservePrice](#)
- [BuyItNowPrice](#)
- All shipping properties ([ShippingDetails](#))

eBay Real Estate Listings

eBay Real Estate listings do not involve legally binding offers to buy and sell, with some exceptions as noted in the eBay online help:

Real Estate Rules

<http://pages.ebay.com/help/policies/real-estate-rules.html>

When you list real property on eBay, you have the option of listing in a regular auction format or in a Classified Ad (advertisement) format (see “[Classified Ad Listings](#)” on page 366). eBay Real Estate auctions enable sellers to advertise their real estate and meet potential buyers. eBay Real Estate is not involved in the actual transaction between buyers and sellers. eBay Real Estate offers two auction options for real estate advertisements, *NonBinding* and *Binding*. The [GetItem](#) function returns a Real Estate auction listing’s current binding option in the [Item.ListingDetails.BindingAuction](#) field.

There are special considerations an application must make when listing items or retrieving eBay Real Estate items on the eBay US and eBay Germany sites:

- All eBay US Real Estate auction and Ad format listings must be listed with Item Specifics. These define many special properties of eBay Real Estate listings. Item Specifics are a set of well-known fields that are common to (or “specific” to) all items in a particular category. For example, in Real Estate categories, Number of Bedrooms and Sq Footage (Sq. ft.) would be Item Specifics. (Fields like Number of Bedrooms wouldn’t make sense for other categories, like books or men’s

- clothing.) See “[Working with Custom Item Specifics](#)” on page 481 for information about Item Specifics.
- Some eBay Germany Ad format categories (e.g., Wohnen) support Item Specifics, and others do not.
 - Real Estate items can only be listed in Real Estate categories. Once an item has been listed in one of the Real Estate categories, you cannot change its meta-category when revising or relisting the item. That is, you cannot change the listing’s category to, say, the Books category instead.
 - Payment method fields are not applicable to Real Estate listings.

In addition to the differences that are applicable for all Real Estate listings, these differences are applicable for Real Estate auction listings:

- eBay US Real Estate auction listings can have durations of 30 days, in addition to standard auction listing durations
- No payment methods ([PaymentMethods](#)) are applicable for Real Estate auction listings.

See “[Classified Ad Listings](#)” on page 366 for differences related to the Classified Ad format.

Digital (Downloadable File) Listings

Important: All digital listings selling on eBay US are required to be listed in a Classified Ad format within the Everything Else > Information Product category. (Other sites do not support digital listings.) If you use the `DigitalDeliveryDetails` field in `AddItem` and related calls, you will receive an error.

To list a downloadable item as a Classified Ad, set the value of `Item.ListingType` to `LeadGeneration` and `Item.ListingSubtype2` to `ClassifiedAd`. Be sure to only list in the Everything Else > Information Product category.

Important: The `ClassifiedAd` enumeration is used to list digital downloads on [ebay.com](#) websites. It cannot be used to list anything on [ebayclassifieds.com](#), which is a different website. The [ebayclassifieds.com](#) site is not supported by this API. For information about the other differences between these two sites, see [Advertising with classified ads on eBay and eBay Classifieds](#).

Using the Classified Ads format, sellers receive a 30-day ad at a fixed price. This solution enables sellers to continue to market their digital goods on eBay; however, because Classified Ad listings are a lead generation tool and do not result in order line items that go through eBay, please be aware that feedback cannot be exchanged between buyer and seller.

All digital delivery fields will be removed from the schema in a future release, and this documentation will be updated with more details at that time.

Note: The remainder of this section is deprecated. It is being maintained in this documentation to help you identify changes you need to make in your application if you are using the previous digital delivery feature.

eBay enables you to list and sell digital files (such as Adobe PDF files) or information (such as a “Word of the Day”) that buyers can access online or have delivered electronically.

To list a digital item via [AddItem](#) and related calls:

- Choose a category that supports digital listings.

Use [GetCategoryFeatures](#) and check the [DigitalDeliveryEnabled](#) flag. (See the [GetCategoryFeatures call reference](#).)

If you list in two categories, both categories must support digital listings.

- In [ListingType](#), specify [FixedPriceItem](#) as the format. (That is, you cannot use an auction-style format.)
- In [DigitalDeliveryDetails](#), provide a file location or alternate delivery method for your digital item. Provide any details buyers need to determine whether they can use the item. Specify the type of file (for example, "MP3 audio file") as well as any special requirements for using the item.

Please note that delivery information will be displayed to a buyer only after payment for the item has been confirmed.

See [AddItem](#) in the eBay Trading API Call Reference for more information about [DigitalDeliveryDetails](#).

- In [Quantity](#), specify the number of copies you would like to sell, up to a maximum of 10,000. You can sell multiple, identical copies of a digital item in the same listing (assuming you are legally authorized to do so).
- In [PaymentMethods](#), only offer PayPal as a payment method so that buyers can access the item immediately once payment is confirmed.

Some sites may require Immediate Payment (see [AutoPay](#)) unless the seller is using Third-Party Checkout. Some sites may require the seller to be VAT registered (see the eBay online help for requirements).

Important: Similar to other items that are not shipped and tracked, digital items are not covered by the PayPal Seller Protection Policy.

- In [PayPalEmailAddress](#), use a PayPal email address that is associated with a PayPal Verified Premier or Verified Business account. This means that you've placed your bank account on file at PayPal and you accept credit card payments from buyers. If you don't have a Verified Premier or Verified Business account, you can use PayPal to get your account Verified and/or upgrade to a Premier or Business account.
- Use [AttributeSetArray](#) to specify whether you are willing to offer a refund if requested by a buyer. Specify the conditions under which you will accept refund requests and the type of refund you will give.

Use [GetAttributesCS](#) to retrieve meta-data that you need in order to specify your refund policy. See "[Offering a Clear Return Policy](#)" on page 223. As the notion of a "return" policy doesn't make sense for digital listings, you can pass in an optional [DigitalDelivery](#) flag to configure [GetAttributesCS](#) to retrieve "refund" policy meta-data instead:

- In the digital delivery variation of the response, there are no changes to the attribute set IDs, attribute IDs, or value IDs. The main difference is that references to "return policy" in the text are replaced with the words "refund policy". This change is cosmetic, and it is intended

only to help the seller understand their options while they are filling in the refund policy details. If a seller specifies that they are listing a digital item, you can render the “refund policy” variation of the applicable meta-data in your Item Specifics or Return Policy form. (You can use the Item Specifics SYI XSL stylesheet to render the data as usual.)

- For digital items, the Item Condition attribute (in Item Specifics) is not applicable in [AttributeSetArray](#), [AttributeArray](#), or [LookupAttributeArray](#). If you pass in the item condition in [AddItem](#), it will be dropped. Therefore, in the digital delivery variation of the [GetAttributesCS](#) response, the Item Condition attribute is also removed from all characteristic sets so that the seller will not be confused by this option.

The [DigitalDelivery](#) flag in the [GetAttributesCS](#) request is a client-side convenience only. If it is too difficult to cache or manage both variations of the [GetAttributesCS](#) meta-data, you can use the existing “return policy” text or post-process the text yourself before presenting it to the seller. Rendering the words “return policy” vs. “refund policy” on the client will have no effect on any information that appears on eBay once the item is listed. For digital items, eBay will display “refund policy” text to buyers. Similarly, regardless of whether you show or hide the Item Condition to the seller, no Item Condition will be included in the listing that appears on eBay.

- In addition, verify that you are legally authorized to sell the digital item when you list it. We recommend that you check eBay’s policies on Downloadable Media before listing your item.

For more information about eBay’s policies regarding digital listings, links to useful pages on the PayPal site, and the latest tips on selling digital items effectively, see the eBay Web site online help:

Selling Digital Items

<http://pages.ebay.com/help/sell/selling-digital-items.html>

Digitally Delivered Goods

<http://pages.ebay.com/help/policies/downloadable.html>

Fields that specify the item location, shipping details, payment methods other than PayPal, gift options, Now and New, and Best Offer are not applicable to digitally delivered items.

You can use [GetItem](#), [GetBidderList](#), and [GetSellerList](#) to retrieve digital listings.

Ticket Listings

This section provides some tips on listing in event ticket categories. This information is mostly relevant to applications that list items.

Note: If you want to search for ticket listings, use the [Finding API](#) and use [aspect filters](#) to refine your search.)

- “[Basic Ticket Listing Requirements](#)” on page 371
- “[Listing Tickets with Product Details](#)” on page 371
- “[Filling in All Ticket Item Specifics Manually](#)” on page 373

Basic Ticket Listing Requirements

Here are the basic requirements for listing tickets on the eBay US and Canada sites:

- The ticket category must be the primary category (if you list in two categories).
- PayPal is a required payment method for listing tickets in some states.

Some states (e.g., New York and Illinois) have ticket resale laws requiring online platforms permitting sales of event tickets to offer full buyer protection. eBay offers full buyer protection on ticket sales to New York and Illinois events through PayPal. eBay therefore requires all sellers to accept PayPal payments for ticket listings when the venue state is New York or Illinois (regardless of where the buyer or seller is located).

- Item specifics ("Number of Tickets" and "Venue State/Province") are required for US and Canada ticket listings.

Item Specifics are a set of well-known fields that are common to (or "specific" to) all items in a particular category. For example, in the Event Tickets category, Event (e.g., "Concert") and Venue (e.g., "Fillmore Theater") would be Item Specifics. (Fields like Event and Venue wouldn't make sense for other categories, like books or men's clothing.)

There are basically two ways to fill in Item Specifics for tickets:

- (*Recommended*) Pre-fill most of the Item Specifics from the tickets catalog, and fill in the Section and Row item specifics yourself. See "[Listing Tickets with Product Details](#)" on page 371.
- If your event is not in the tickets catalog, fill in the required and optional Item Specifics yourself. See "[Filling in All Ticket Item Specifics Manually](#)" on page 373.

On the US and CA sites, tickets are listed the Sports, Concerts, Theater, and Other categories. Call [GetCategories](#) to determine the category IDs.

Listing Tickets with Product Details

eBay offers a tickets catalog that provides stock information, including a venue map, for most tickets available in the eBay US and Canada marketplaces.

["Allow eBay to Find and Add Ticket Details from the Catalog For You"](#) on page 371
["Find Ticket Details by Searching the Catalog Yourself"](#) on page 372

Allow eBay to Find and Add Ticket Details from the Catalog For You

The easiest way to list a ticket is to specify the event title, venue, date, and time in [ProductListingDetails.TicketListingDetails](#) in [AddItem](#).

This approach is useful when you do not want to first search the ticket catalog to determine the product ID for a stock product. This approach helps sellers list tickets quickly, and it requires fewer API calls than the search-to-sell flow (i.e., finding ticket details by searching the catalog).

For ticket listings, the catalog-enabled event tickets category must be the primary category.

When you list a ticket in this manner, eBay searches the ticket catalog and (assuming a unique match is found) pre-fills certain Item Specifics and other data based on the catalog data. If a single matching product is found, eBay lists the item with that product's details, including a stock venue map (if available) and other stock information. The stock venue map will be used as the listing's picture and Gallery image. Typically, the venue map shows the venue's seating chart relative to the stage, screen, or field.

If no matches are found, the call returns an error and the item is not listed. If this occurs, use one of these approaches to correct the problem:

- Check to make sure you did not specify extra words in the event or venue name, and check to make sure the date and time are correct (e.g., if the ticket says 7:05 PM, make sure you did not pass in 7:00 PM).
- Search for an EPID (see “[Find Ticket Details by Searching the Catalog Yourself](#)” on page 372). If it is, use the eBay product ID to pre-fill the ticket data.
- If the ticket is not in the catalog, fill in the Item Specifics and do not pass in [ProductListingDetails](#).

If more than one match is found, the call returns an error (indicating that too many products were found) and a list of product IDs for the products that matched the keywords you passed in. If this occurs, try using more keywords (such as the event time, if you did not specify it), or search for a specific EPID (see “[Find Ticket Details by Searching the Catalog Yourself](#)” on page 372).

If you pass in an old product ID (instead of ticket keywords) and only one match is found, eBay will list the item with the latest version of the product and the latest product ID, and the call will return a warning indicating that the data has changed.

If you pre-fill most Item Specifics from the ticket catalog, you need to also pass in custom item specifics that provide the Venue State/Province and Number of Tickets. Other required item specifics will be pre-filled from the catalog.

It is also a good idea to pass optional Item Specifics that are not in the catalog (such as the section and row), to help buyers more quickly discern which ticket they are purchasing.

Call [GetCategorySpecifics](#) for a list of the custom Item Specifics that are supported and required in the category you’re listing in.

Find Ticket Details by Searching the Catalog Yourself

If the seller wants to search for and preview event details in the catalog before including them in a listing, you can search for the event, view matching products, select one, and pass its eBay product ID (EPID) in [ProductListingDetails](#) in [AddItem](#).

This approach involves more steps than using [TicketListingDetails](#) (described above), but it enables previewing an event before including it in your listing. It also may serve as a workaround if eBay can't find a matching product when you use [TicketListingDetails](#).

Use [FindProducts](#) in the Shopping API (see the [FindProducts documentation](#)). Pass keywords describing the ticket in [QueryKeywords](#), and then choose a product from the results. Pass the product ID you found in [ProductListingDetails.ProductReferenceID](#) in [AddItem](#).

As stated in the prior section, if you pre-fill most Item Specifics from the ticket catalog, you need to also pass in custom item specifics that provide the Venue State/Province and Number of Tickets. Other required item specifics will be pre-filled from the catalog.

It is also a good idea to pass optional Item Specifics that are not in the catalog (such as the section and row), to help buyers more quickly discern which ticket they are purchasing.

Call [GetCategorySpecifics](#) for a list of the custom Item Specifics that are supported and required in the category you're listing in..

Filling in All Ticket Item Specifics Manually

If there is no product available that matches your ticket, use item specifics to specify the details.

Ticket listings have certain required Item Specifics. It is also a good idea to pass optional Item Specifics that are not in the catalog (such as the section and row).

Call [GetCategorySpecifics](#) for a list of the custom Item Specifics that are supported and required in the category you're listing in.

For example, you must manually include item specifics for "Number of Tickets" and "Venue State/Province" yourself (because they're required). In addition, we strongly recommend that you specify "Event Date" when you fill in your own item specifics. Use the listing title for the event name.

Regardless of whether you pre-fill required item specifics based on a catalog product or fill them in yourself, we also strongly recommend that you always add the optional "Section" and "Row" item specifics. Most ticket buyers filter and sort tickets based on these item specifics. This means listings that are missing these item specifics typically appear after all other listings in search results. Also, eBay's venue map functionality leverages the Section item specific that the seller specifies.

[GetCategoryFeatures](#) and [GetCategorySpecifics](#) return details about custom item specifics for tickets. For example:

- Item specifics recommendations indicate that "Number of Tickets" and "Venue State/Province" are required. (That is, you need to pass them in [AddItem](#) if you don't pre-fill them from a catalog product).
- You use the "Type" item specific to clarify the event type (e.g., Type="Baseball").
- You use the "Event Date" item specific to indicate the date of the event.

Part IV

Managing Listed Items

Managing Item Listings	375
Communication Between Members	402
Completing the Sale	415
Checkout	407
Leaving and Retrieving Feedback	432
Unpaid Item Disputes	455
Buyer Disputes	476

Chapter 19

Managing Item Listings

This chapter discusses the means for managing item listings. This includes relisting an item, revising an item, and retrieving seller events. For information about using the API to submit a bid (or purchase) on an item, see “[PlaceOffer](#)” on page 904.

- “[Relisting Items](#)” on page 376
- “[Revising Items](#)” on page 381
- “[Changing Item Descriptions](#)” on page 388
- “[Ending Items Early](#)” on page 389
- “[Working with Seller Events](#)” on page 391
- “[Learning About Bidders](#)” on page 393
- “[Putting Store Items on Sale](#)” on page 394
- “[Using Selling Manager Calls in the Trading API](#)” on page 397

For useful tips on revising items on various sites, also see these Knowledge Base articles:

Rules for Revising Items

https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=732

ReviselItem and RelistItem Best Practices and Version Concerns

https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=267

The Concept of Relisting

Item listings that end are typically eligible to be listed again. Listing an item under these conditions is known as *relisting*. See the online help on the eBay site for specific information on when an item is and is not eligible to be relisted without fees.

<http://pages.ebay.com/help/sell/relist.html>

When a seller relists an item, it can be done without any changes to the item. Just specify the item ID of the original listing that ended.

If desired, some parts of the item’s definition can be changed. For example, the seller might specify a lower price for the item, hoping to better entice a user to purchase the item. Not all elements of an item’s definition can be changed on relisting, and some changes can cause fees to be incurred where the relist would otherwise be free. For instance, if the seller relists an item and specifies a fee-based listing enhancement (where the original item did not use that listing enhancement), then the seller would pay more for it.

Note: For Best Match information related to multi-variation listings, see the Best Match information at the following topic:
[Multi-quantity Fixed Price listings with variations](#).

See “[Relisting Items](#)” on page 376 for details. For information on [RelistItem](#), see the chapter “[RelistItem](#)” on page 906. Refer to the [eBay Trading API Call Reference](#) for descriptions of the call-specific input and output fields.

The Concept of Revising

An item that is still active on the eBay site may be changed by the seller. This is known as *revising* an item. When an item is revised, the seller specifies a new value for one or multiple elements of the item’s definition. This could entail supplying a new, non-blank value where there was no such value for the item as originally listed. It could also mean specifying a blank value where there was a non-blank value. Or it could mean specifying a value of false where there was previously a value of true (as in to turn off some option, like seller allowing the buyer to change the payment at checkout). There are certain rules that govern what can be changed when a listing has bids or is within 12 hours of the end of the listing.

Note: For Best Match information related to multi-variation listings, see the Best Match information at the following topic: [Multi-quantity Fixed Price listings with variations](#).

See “[Revising Items](#)” on page 381 for details. For information on [ReviselItem](#), see the chapter “[ReviselItem](#)” on page 918. Refer to the [eBay Trading API Call Reference](#) for descriptions of the call-specific input and output fields.

Events Related to a Listing

A seller may be interested in various events related to an in-progress listing. One such an event is a change in the current price of a competitive-bidding listing (indicating someone has bid on the item or placed a higher bid). Another is the listing reaching the end of its duration (i.e., the end of the listing). An application can use [GetSellerEvents](#) to determine when such events have occurred and then retrieve the items for which these events have occurred and present them to the seller. See “[Working with Seller Events](#)” on page 391. Also, for information on [GetSellerEvents](#), see “[GetSellerEvents](#)” on page 858.

Relisting Items

- “[General Steps](#)” on page 376
- “[Indicating Changed Fields When Relisting an Item](#)” on page 377
- “[Removing a Field When Relisting an Item](#)” on page 377
- “[Changing Payment Methods in a Relisted Item](#)” on page 378
- “[Relisting Items with Product Details \(Catalogs\)](#)” on page 379
- “[Relisting and Versions](#)” on page 380

General Steps

If an item’s listing ends, it may be eligible for relist.

See the online help on the eBay site for specific information on when an item is and is not eligible to be relisted without fees.

<http://pages.ebay.com/help/sell/relist.html>

A relist creates a new listing with all of the properties of the old one, except those the seller explicitly changes.

To relist an item, use **RelistItem**. Doing this involves three general steps: set up the execution environment, specify the item to relist (by its item ID), and make the API call. Relisting an item entails creating and specifying an **ItemType** object representing the item to be re-listed and specifying which item fields (if any) are being changed in the relist. Items generally can be relisted without changing any values.

For information about considering an item's rank in search results, see “[Considering Best Match When Revising and Relisting Items](#)” on page 184.

The application must assign to the **Item.ItemID** property the original item ID (identifying the item being relisted), and specify any changes to be made with the relist. The item ID must be specified for every **RelistItem** attempt, whether or not there are changes to the item's definition values.

If no changes are made to the item definition with the relist, the only value assigned to this **Item** object is the item ID. This object is later assigned to the **Item** property of the **RelistItem** requestobject.

If there *are* changes to the item definition, the corresponding **ItemType** object properties must be assigned the new values. Populate the properties of this **ItemType** object *only* with the item ID and changed values.

If the seller changes the listing category, data that depends on the category is still required to be consistent with the category the seller has selected. This means the seller may need to change the listing details when they change the category. For example, the seller may need to change the listing's Item Specifics or product details to be consistent with the new category (see “[Relisting Items with Product Details \(Catalogs\)](#)” on page 379 for more information about this particular case).

The category requirements for providing safe payment methods apply to a relisted item the same way they applied to the original item.

The **CategoryMappingAllowed** flag is transient. If it was specified as **true** when the item was originally listed, it will not automatically be set to true when you relist the item. Therefore, it should be specified again as a new property. See “[Mapping Old Category IDs to Current IDs](#)” on page 163.

Indicating Changed Fields When Relisting an Item

To change an item property when using **RelistItem**, the application simply specifies the new value of the property in the **Item** object used as input. For an example, see “[Sample XML for Modifying and Removing Fields with a Relist](#)” on page 378.

Removing a Field When Relisting an Item

You can remove most fields when you relist an item. However, there are some fields you cannot remove or change when you relist an item (such as the listing site).

In some cases, you can remove a setting from a listing by changing the value of the corresponding field. For example, you can set a boolean like **PrivateListing** to false to “remove” the private listing setting.

However, in several cases, you need to use a special mechanism to remove a property from a listing. This is particularly true for strings (because an empty string is not the same as a deleted string), monetary amounts (because a price of 0.00 usually means “free” rather than “unspecified”), and other fields where null values may be ambiguous. To delete an item property of this kind, use **DeletedField** to specify the property. The documentation of **DeletedField** lists the fields that are subject to this rule.

For example, if you are changing the start price and the Listing Designer theme for the item and removing the subtitle, your request should contain the following XML:

Example 19-1 Sample XML for Modifying and Removing Fields with a Relist

```
...
<Item>
  <ItemID>4046156496</ItemID>
  <StartPrice currencyID="USD">200.0</StartPrice>
  <ListingDesigner>
    <ThemeID>10</ThemeID>
  </ListingDesigner>
</Item>
<DeletedField>Item.SubTitle</DeletedField>
...
```

If you are deleting a listing enhancement (that is, deleting a value found in **ListingEnhancementsCodeType**), an example of the relevant XML is the following. Note that if, in the relist, you accidentally include both a listing enhancement and a deletion of the enhancement (not shown in example), the enhancement you supply overrides your deletion.

Example 19-2 Sample XML for Deleting a Listing Enhancement with a Relist

```
...
<Item>
  <ItemID>4046156496</ItemID>
  <StartPrice currencyID="USD">200.0</StartPrice>
</Item>
<DeletedField>Item.ListingEnhancement[BoldTitle]</DeletedField>
...
```

If the item being relisted had a **ReservePrice**, when you relist it this value must be equal to or less than the value of the **ReservePrice** in the *original* listing.

Changing Payment Methods in a Relisted Item

A listing must have at least one valid payment method. To determine the payment methods that each eBay site supports, see “[Payment Method Differences \(PaymentMethod\)](#)” on page 601.

When you revise a listing and you specify a payment method that is invalid for the target site, eBay ignores the invalid payment method, lists the item, and returns a warning to indicate that the invalid payment method was ignored. If multiple payment methods were invalid, the warning indicates that they were all ignored.

If you modify the listing so that it includes no valid payment methods, an error is returned. This situation could occur when the seller removes all valid payment methods or when all the payment methods specified for the item are no longer valid on the target site.

See also “[Changing Payment Methods When Revising an Item](#)” on page 384.

Relisting Items with Product Details (Catalogs)

Items in certain categories can have details whose values are well known and don't vary from listing to listing. Instead of filling in this standard information for every listing, a seller can create a listing in which this information has already been partially filled out (pre-filled). This data is called product details.

If an item is listed with product details and you change the relevant category, you need to make sure the new category is mapped to the characteristics set associated with the product. The product data must always be consistent with at least one of the listing's categories. If the item already includes product details, but that information is inconsistent with the new category, you must remove the product information or replace it with new product information that is consistent.

These are the common scenarios when re-listing an item with product details:

- Add product details to the listing – to add product details, set **Item.ProductListingDetails** to contain all product information that you want to include in the listing. Specify the ID of the product that should be used to pre-fill the listing, and use the optional child elements to specify whether you want the listing to include a stock photo, whether you want to use the stock photo as the item's gallery image, and whether to include a stock summary in addition to your own item description.
- If you choose to include a stock photo in the listing, you must remove **GalleryURL**. Instead, you can either use the stock photo as the Gallery image (**UseStockPhotoURLAsGallery**) or use the first image passed in **PictureURL** (if any) as the Gallery image. See “[Using a Stock Photo in an Item Listing](#)” on page 210.
- Remove all existing product data (including Item Specifics) – to remove all existing product data from the listing, specify **Item.ProductListingDetails** in **DeletedField** and **Item.ItemSpecifics** in **DeletedField**.

If you remove product data without removing the item specifics, it has the following effects:

- Pre-filled item specifics are removed from the listing.
 - Item specifics that are not pre-filled are maintained (unless you remove them or change the category to one that does not support the previously specified item specifics).
 - The stock photo (if any) is removed from the listing (and from the Gallery, if applicable).
 - The link to additional summary information is removed from the listing.
- Replace previous product data (including pre-filled Item Specifics) – to replace the previous product data with data for a different product, set **Item.ProductListingDetails** to contain the new product data. To include optional values, set the **Item.ItemSpecifics** property to contain the additional product details that you want to include.

After executing **RelistItem** with product details, you should compare the response to the request to make sure that the data that was actually listed matches the data that you sent.

Removing a Listing Subtitle When Relisting an Item

You can remove certain properties from a listing, such as the subtitle, when you relist an item. To remove a subtitle, specify **Item.SubTitle** in **DeletedField**.

Relisting and Versions

When relisting, you may get an error about certain elements not being specified. It is possible for this to occur when an item is initially listed with an older version and then relisted with a newer version.

There may be little or no mapping from older tags to newer tags and thus the old listing does not contain enough information to be successfully relisted with the newer version (for example, details related to shipping or to item specifics).

One solution would be to relist the item with the older version. However, that requires knowing what version is best, and you might not have been the one who originally listed the item.

Troubleshooting tips:

- Try relisting the item with a lower version.
- View or even relist the item via eBay Web site; sometimes the site provides information not available via API, information that might give insights into a possible solution.

Relisting Eligible Administratively Cancelled Items

You can use **RelistItem** to quickly relist an item that was administratively cancelled due to a violation of an eBay listing policy, for certain violations (see <http://pages.ebay.com/help/policies/listing-ov.html> for more information on eBay's listing policies). Using **RelistItem** in this case is more efficient than using **AddItem**. In some cases, sellers may even be able to relist partially sold multi-quantity listings which have been prematurely removed from the site depending upon the violation.

You can use either **GetItem** or **GetSellerList** to determine if your item has been administratively cancelled and can be relisted. Specifically, if the **GetItem** or **GetSellerList** response contains **Item.SellingStatus.AdminEnded** as **true**, the item can be relisted.

Note that **GetItem** or **GetSellerList** response will include **Item.ItemPolicyViolation.PolicyID** and **Item.ItemPolicyViolation.PolicyText**, which provide the ID and a brief textual description, respectively, of the violated policy. **GetItem** and **GetSellerList** will only return information about administratively cancelled listings to the seller of the item.

When using **GetSellerList**, if you are the seller and specify **AdminEndedItemsOnly** as true in the request, all other specified filters are ignored and only admin-canceled listings are contained in the response, if any.

These enhancements apply to all eBay sites (including eBay Motors but not Half.com).

Using VerifyRelistItem to test RelistItem

An application can also test the listing of an item by using [VerifyRelistItem](#) instead of [RelistItem](#). [VerifyRelistItem](#) uses the same inputs (and in the same ways) and returns the same values as [RelistItem](#), except that it does not actually relist the item. An application could thus use [VerifyRelistItem](#) to test the item definition for validity and to present the seller with a preview of the estimated relisting fees before the item is actually relisted. Once tested, the application could use the same [ItemType](#) object that was used with [VerifyRelistItem](#) and use it again, unchanged, with [RelistItem](#).

For information on [RelistItem](#), including executing the call and business logic, see the chapter [RelistItem](#). For information on [VerifyRelistItem](#), see the chapter, [VerifyRelistItem](#).

Check for a Message node in the [VerifyRelistItem](#) response because it may indicate that a [RelistItem](#) attempt will be unsuccessful; see the following Knowledge Base article, based on [AddItem](#) and [VerifyAddItem](#):

VerifyAddItem Returns Success with No Errors, but AddItem Returns Failure with Errors
https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=912

Refer to the [VerifyRelistItem](#) page in the Call Reference for more information.

Revising Items

- ["Qualifications to Revise an Item" on page 382](#)
- ["Using the VerifyOnly Field to Test Your Changes" on page 382](#)
- ["Item Changes Allowed During Revisions" on page 382](#)
- ["Revising eBay Giving Works \(Charity\) Listings" on page 383](#)
- ["Revising eBay Stores Listings" on page 383](#)
- ["Revising Items After Purchases are Made" on page 383](#)
- ["Removing a Field When Revising an Item" on page 383](#)
- ["Changing Payment Methods When Revising an Item" on page 384](#)
- ["Revising Multiple-Choice \(Unbounded\) Fields" on page 384](#)
- ["Revising US eBay Motors Listings" on page 385](#)
- ["Revising Real Estate Listings" on page 386](#)
- ["Revising Items with Product Details \(Catalogs\)" on page 386](#)
- ["Changing a Listing Start Price" on page 388](#)
- ["Reducing Reserve Price" on page 388](#)

Sellers often need to change a currently active listing. Typically, the change makes the item more appealing to buyers.

Changing the properties that define an active listing is known as *revising* the item. This is accomplished using [ReviselItem](#). This call is directly analogous to the Revise Your Item (RYI) page on the eBay site.

When a seller revises an item, the seller might add information that was not in the original listing, such as information that answers a question received by the seller. See also "[Listing Policies](#)" on page 174.

For information about considering an item's rank in search results, see "[Considering Best Match When Revising and Relisting Items](#)" on page 184.

If you are only changing a listing's price or quantity, consider using `ReviselInventoryStatus`, which is a faster (lighter-weight) call than `ReviselItem` or `ReviseFixedPriceItem`.

Revising an item using `ReviselItem` or `ReviseFixedPriceItem` entails these basic steps:

- 1 Specify the item to revise (by its item ID) in `Item.ItemID`.
- 2 Specify the item fields to add or change.
 - Not all fields can be revised. See "[Item Changes Allowed During Revisions](#)" on page 382.
 - If you are deleting a field, see "[Removing a Field When Revising an Item](#)" on page 383.

Qualifications to Revise an Item

To revise an item listing using `ReviselItem`, the following conditions must be met:

- The requester is the seller. Users cannot revise others' items.
- The item listing must still be active. It is not possible to revise an ended item listing. (But it might be eligible to be relisted, with property changes. See "[Relisting Items](#)" on page 376.)

Additionally, there are restrictions on what can be changed based on whether the listing has bids and based on whether the listing will end within twelve hours. For more on this, see "[Item Changes Allowed During Revisions](#)" on page 382

Using the VerifyOnly Field to Test Your Changes

You can use the `VerifyOnly` boolean field with the flag set to 'true' in `ReviselItem`, if you want the response to include the calculated listing price change (if there is an increase in the BIN/Start price) but you do not want the values to persist in the database.

Including the `VerifyOnly` field in `ReviselItem` is similar to using `VerifyAddlItem` or `VerifyRelistItem` to test calls before you actually add or relist the item.

Item Changes Allowed During Revisions

Making changes to an item's listing using `ReviselItem` is subject to a few restrictions, depending on whether any bids have been placed on the item, and whether the change is being attempted within twelve hours of the end of the auction.

In general, you cannot change the listing format (e.g., auction vs. fixed-price) of any listing.

See the *Restrictions on Revising Listings* help page for a complete list of what can be changed for auction, fixed-price, and classified ad listings.

Restrictions on Revising Listings
http://pages.ebay.com/help/sell/revising_restrictions.html

The next few subsections of this chapter provide details that you should understand when revising certain features via the API, such as item specifics, eBay Motors details, payment methods, and more. Additional information may be provided in other sections of this guide, where applicable.

For a list of the fields that can be modified in the API, see [ReviselItem](#) and [ReviseFixedPricelItem](#) in the *eBay Trading API Call Reference*.

If you change the category, please note that the [CategoryMappingAllowed](#) flag is transient. If it was specified as `true` when the item was originally listed, it will not automatically be set to true when you revise the item. Therefore, it should be specified again as a new property. See “[Mapping Old Category IDs to Current IDs](#)” on page 163.

Revising eBay Giving Works (Charity) Listings

You cannot use [ReviselItem](#) or [ReviseFixedPricelItem](#) to modify eBay Giving Works items. See “[Identifying Listings that Benefit Nonprofits](#)” on page 234 for additional information.

Revising eBay Stores Listings

eBay no longer distinguishes between Store Inventory and regular fixed-price listings. The rules for revising items in your eBay Store are the same as the rules for revising any fixed-price listings. See “[Item Changes Allowed During Revisions](#)” on page 382.

Revising Items After Purchases are Made

You can revise a multi-quantity fixed-price listing after buyers have purchased items from it. In these cases, [GetItem](#) normally returns the latest (revised) details of the listing.

If a buyer purchases items from a multi-quantity fixed-price listing, eBay saves a snapshot of the listing as it was at the time of the purchase. The [TransactionID](#) of the order is associated with that snapshot. Sellers and buyers can retrieve the snapshot of the listing by calling [GetItem](#) and passing [TransactionID](#) in the request. (The snapshot is also available in [GetItemTransactions](#) and related calls.)

Sellers of an item are allowed to view any purchase snapshots of the listing (any transaction or order), whereas buyers can only view the purchase snapshots for which they are the successful buyer. Order line items (purchases) can be viewed for up to 90 days from the creation date.

If you are using [Notifications](#), you can set the order line item event to trigger your notifications or use an event related to the entire listing. Be aware that if you're using item purchase snapshot data, fields like [ListingStatus](#) may report an [Active](#) status (because the listing was [Active](#) on the order line item (transaction) date), but the listing may actually have ended.

Removing a Field When Revising an Item

You can remove certain fields when you revise an item, subject to the rules described in “[Item Changes Allowed During Revisions](#)” on page 382 and in the [ReviselItem](#) field descriptions in the *eBay Trading API Call Reference*.

In some cases, you can remove a setting from a listing by changing the value of the corresponding field. For example, you can set a boolean like [PrivateListing](#) to `false` to “remove” the private listing setting.

However, in several cases, you need to use a special mechanism to remove a property from a listing. This is particularly true for strings (because an empty string is not the same as a deleted string), monetary amounts (because a price of 0.00 usually means “free” rather than “unspecified”), and other fields where null values may be ambiguous. To delete an item property of this kind, use **DeletedField** to specify the property. The documentation of **DeletedField** lists the fields that are subject to this rule.

For example, if you are changing the start price and the Listing Designer theme for the item and removing payment instructions, your request should contain the following XML:

Example 19-3 Sample XML for Modifying and Removing Fields with a ReviseItem

```
...
<Item>
  <ItemID>4046156497</ItemID>
  <StartPrice currencyID="USD">200.0</StartPrice>
  <ListingDesigner>
    <ThemeID>10</ThemeID>
  </ListingDesigner>
</Item>
<DeletedField>Item.ShippingDetails.PaymentInstructions</DeletedField>
...
```

Changing Payment Methods When Revising an Item

A listing must have at least one valid payment method, but you can change it when you revise the item.

Changing a payment method for a revised item involves adding the change to an **ItemType** object. For example, if you want to change the payment method for an item currently listed, you would follow these steps, after authenticating and creating the request object:

- Create an **ItemType** object and set the item ID of the item you want to change.
- Set the **ItemType** object’s **PaymentMethods** property.
- Add the **ItemType** object to the request.
- Create a request object for the call you want to use (for example, **ReviseItemRequestType**).
- Create a response object (for example, **ReviseItemResponseType**). Make the request and store its result set in the response.

Parse the response to obtain the values you need.

Revising Multiple-Choice (Unbounded) Fields

Because **ReviseItem** replaces only item properties that are included in the input **ItemType** object and list of properties changed, be careful when updating properties that are made up of multiple choice options, such as shipping option. In these cases, specify all intended options. For instance, if the listing already has options #1 and #2 and the revision change is to add option #3, include all three in the property of the input **ItemType** object.

Revising US eBay Motors Listings

In addition to the normal rules for revising listings, the following rules apply for US eBay Motors listings:

- **Site:** Set `Item.Site` to the `eBayMotors` site code and specify "100" as the site ID in the request URL.
- **PrimaryCategory** and **SecondaryCategory**: You cannot change the *meta-category* of a vehicle. For example, you cannot change a truck to a motorcycle.

However, you can change a subcategory in which a vehicle is listed, as long as there are no bids on the listing. For example, you can change the primary or secondary category from a motorcycle subcategory to another motorcycle subcategory.
- **ListingDesigner:** Listing Designer elements are ignored when revising any listings (vehicular and non-vehicular) on the US eBay Motors site.
- **ShippingDetails:** You cannot revise shipping, tax, and insurance elements for any listings (vehicular and non-vehicular) on the US eBay Motors site.
- **BuyItNowPrice:** The rules for revising the Buy It Now Price for eBay Motors listings are the same as for other listings, except for the following. You can use the `BuyItNowPrice` argument to add or lower the Buy It Now Price of a vehicle listing if all of the following conditions exist:
 - The new Buy It Now Price must be at least \$1.00 less than the original Buy It Now Price.
 - The new Buy It Now Price must be at least \$1.00 greater than the current high bid price.
 - The Buy It Now Price can be lowered regardless of the amount of time remaining in the listing.
- **ReservePrice:** The reserve price for US eBay Motors vehicle listings can be lowered or removed regardless of the amount of time remaining in the listing. See "[Reducing Reserve Price](#)" on page 388.

Note: If you relist this item, the `ReservePrice` value must be equal to or less than the value of the `ReservePrice` in the *original* listing.

If the item has no bids, set `ReservePrice` to any value within this range:

- Lowest: \$0.00 (to remove the reserve price)
- Highest: `ReservePrice` - 1 (\$1.00 below the current reserve price)

If the item has bids, set `ReservePrice` to a value within this range:

- Lowest: `CurrentPrice` + 1 (\$1.00 above the current high bid)
- Highest: `ReservePrice` - 1 (\$1.00 below the current reserve price)

If the vehicle listing has bids, then users who view the listing will see a message indicating that the next bid will meet the reserve price. See the Fees Overview on the eBay Web site for information about fee credits that may be available when the reserve price is removed.

Fees Overview

<http://pages.ebay.com/help/sell/fees.html>

When retrieving items from the US eBay Motors site, `GetItem` returns information about changes to Buy It Now and reserve prices in the `ReviseStatus` property of the `Item` object.

Revising Real Estate Listings

Your application can list real estate to the eBay US and Germany sites (`SiteCodeType` values `US` and `Germany`, respectively).

eBay Real Estate listings do not involve legally binding offers to buy and sell, with some exceptions as noted in the eBay online help:

<http://pages.ebay.com/help/policies/real-estate-rules.html>

When you list real property on the US eBay site, you have the option of listing in an auction format or in an Ad format (`ListingTypeCodeType` values `Chinese` and `LeadGeneration`, respectively):

- eBay Real Estate listings enable sellers to advertise their real estate and meet potential buyers. eBay Real Estate is not involved in the actual transaction between buyers and sellers. eBay Real Estate offers two auction formats for real estate advertisements, binding and non-binding. Calls such as `GetSellerList` and `GetItem` return a Real Estate auction listing's current format in the `Item.ListingDetails.BindingAuction` property (which returns a value of true if the listing is binding).
- eBay Real Estate also offers sellers the option of advertising their property in an "Ad" format rather than an auction format. When the ad format listing (`ListingTypeCodeType` value `LeadGeneration`) is used, no auction or sale of the real estate takes place. Instead, a user can leave a "lead" for the item to show interest. An application can retrieve these leads for the seller using the `GetAdFormatLeads` call. To list a property in the Ad format, set the value of the `Item.ListingType` property to the `Ad` type using `AddItem`.

When you list real property on the eBay Germany site, you have the option of listing in the Ad format. As with the US site, when the ad format is used, no auction takes place. Instead, a user can leave a lead for the item listing to show interest. An application can retrieve these leads for the seller by calling `GetAdFormatLeads`.

When you use the API to list real estate, certain standard inputs (such as payment options) are not applicable. Also, the choice of possible values for some inputs (such as the listing's duration) are different for real estate items than for other types of listings. Please see "[eBay Real Estate Listings](#)" on page 367 for a list of these differences.

Revising Items with Product Details (Catalogs)

Items in certain categories can have details whose values are well known and don't vary from listing to listing. Instead of filling in this standard information for every listing, a seller can create a listing in which this information has already been partially filled out (pre-filled). This data is called product details.

When an auction item has bids or the listing ends in fewer than 12 hours, you cannot change the product ID and you cannot remove the product from the listing. However, you can change the values of the `IncludeStockPhotoURL`, `UseStockPhotoURLAsGallery`, and `IncludePrefilledItemInformation` flags. (This restriction does not apply to fixed-price listings.)

When an auction item has no bids and does not end within 12 hours, you can use [ReviselItem](#) to add, modify, or remove product details.

If an item is listed with product details and you change the relevant category, you need to make sure the new category is mapped to the characteristics set associated with the product. The product data must always be consistent with at least one of the listing's categories. If the item already includes product details, but that information is inconsistent with the new category, you must remove the product information or replace it with new product information that is consistent.

These are the common scenarios when revising an item with product details:

- Add product details to the listing – to add product details, set [Item.ProductListingDetails](#) to contain all product information that you want to include in the listing. Specify the ID of the product that should be used to pre-fill the listing, and use the optional child elements to specify whether you want the listing to include a stock photo, whether you want to use the stock photo as the item's gallery image, and whether to include a stock summary in addition to your own item description.
- If you choose to include a stock photo in the listing, you must remove [GalleryURL](#). Instead, you can either use the stock photo as the Gallery image ([UseStockPhotoURLAsGallery](#)) or use the first image passed in [PictureURL](#) (if any) as the Gallery image. See “[Using an eBay Stock Photo as a Gallery Picture](#)” on page 214.
- Remove all existing product data (including Item Specifics) – to remove all existing product data from the listing, specify [Item.ProductListingDetails](#) in [DeletedField](#) and [Item.ItemSpecifics](#) in [DeletedField](#).

If you remove product data, it has the following effects:

- Pre-filled item specifics are removed from the listing.
 - Item specifics that are not pre-filled are maintained (unless you remove them or change the category to one that does not support the previously specified item specifics).
 - The stock photo (if any) is removed from the listing (and from the Gallery, if applicable).
 - The link to additional summary information is removed from the listing.
- Replace previous product data (including pre-filled Item Specifics) – to replace the previous product data with data for a different product, set [Item.ProductListingDetails](#) to contain the new product data.
- Add, modify, or change item specifics that were not pre-filled – to add, modify, or change only the item specifics that were not pre-filled, see “[Revising and Relisting with Custom Item Specifics](#)” on page 489 for rules on revising Item Specifics.

Changes to product data in the system do not affect an existing listing. Therefore, the product ID and any pre-filled data that was present at the time of the [AddItem](#) request remains constant at the time of the [ReviselItem](#) request.

After executing [ReviselItem](#) with product details, you should compare the response to the request to make sure that the data that was actually listed matches the data that you sent. For example, if you changed the value of an item specific but the value is now pre-filled in the catalog, the pre-filled value overrides the value you send in.

Changing a Listing Start Price

When you revise an item, you can change the listing start price. However, if you reduce the start price, the insertion fee is not credited (for the US site and other participating sites). The insertion fee is based on the maximum price the item has had in its life cycle.

Reducing Reserve Price

Sellers can lower Reserve Prices on listings on the US eBay site, eBay UK site, or eBay Motors site, even after bids have been placed, to increase the likeliness of their items selling. To determine whether a site or category supports this capability, call [GetCategories](#) and check the [ReduceReserveAllowed](#) and [ORRA](#) fields. See “[Determining Whether a Category Supports Reserve Prices](#)” on page 169. Depending on the site and category, you also may need to call [GetCategoryFeatures](#) and check [MinimumReservePrice](#).

When a category supports lowering the reserve, a seller can adjust the reserve price through the Revise Your Item process or by using [Reviseltem](#), as long as no bids have exceeded the original reserve price. For most categories, this is only allowed when 12 or more hours remain for the listing. The reserve price can be reduced downwards to a minimum of one bid increment above the listing's current price. Once the reserve has been lowered, bidders are notified of the change via email, My eBay, and the listing itself.

If the item has no bids, set [ReservePrice](#) to any value within this range:

- Lowest: 0.0 (if the category supports removing the reserve)
- Highest: [ReservePrice](#) - bid increment

If the item has bids, set [ReservePrice](#) to a value within this range:

- Lowest: [CurrentPrice](#) + bid increment
- Highest: [ReservePrice](#) - bid increment

If the new reserve price is lower than the high bidder's proxy bid, the proxy bid is also reduced to one increment below the new bid price, to prevent bidders from unexpectedly becoming winning bidders when their prior bid had not yet met the Reserve price. For example, imagine a US listing starts at \$1 with a reserve of \$100, and the first bidder bids \$75. The current price stays at \$1 since the bid didn't meet the reserve. Then, if the seller then reduces reserve to \$50, the bidder's proxy bid is reduced to one bid increment below \$50. For a chance at becoming a winning bidder, the bidder would need to place another bid.

No credit is given for Reserve Price Fee if the listing is revised to reduce the Reserve Price.

See also “[Revising US eBay Motors Listings](#)” on page 385.

Changing Item Descriptions

Information often must be added to an item description, and in some cases information must be replaced.

New information might be a response to a buyer's question that all potential buyers should know. Or new information could be a logo added by a listing application.

The following calls can be used for this purpose: [ReviselItem](#) (see "Revising Items" on page 381 and "ReviseItem" on page 918) and [AddToltemDescription](#) (see "AddToItemDescription" on page 789).

When you use [ReviselItem](#) to add or change description information, specify a value of [Append](#) (or another [DescriptionReviseModeCodeType](#) value such as [Prepend](#) or [Replace](#)) in [Item.DescriptionReviseMode](#). Also include the new text in [Item.Description](#), and as usual specify the Item ID in [ItemType.ItemID](#).

The purpose of [AddToltemDescription](#) is to add information to the description of an item. [AddToltemDescription](#) appends text to the end (the bottom) of the text already in the description, leaving the existing text as it was. To change the description in other ways, you must revise the item using [ReviselItem](#) (see "Revising Items" on page 381 and "ReviseItem" on page 918).

To use [AddToltemDescription](#) to add text to a listing's description, the following conditions must be met:

- The requesting user making the call must be the item's seller. (Users are not allowed to modify other users' listings.)
- The listing for which the description is being added to must be an active listing. An application cannot make this change to an ended listing.
- The item cannot be an eBay Giving Works listing (for information about these type listings, see "Identifying Listings that Benefit Nonprofits" on page 234).

The application must pass the item ID that uniquely identifies the item listing to be changed. Pass the item ID in the [ItemID](#) property (which is of type [ItemIDType](#)) of the call's request object. The application must also specify the text that is to be appended to that listing's description. Set the value of the request object's [Description](#) property (type [string](#)) to the text to be appended. Both of these values are required. The example below shows [AddToltemDescription](#) being used to add the string "This MP3 player is only six months old" to the end of the item [2202546967](#).

[AddToltemDescription](#) does not return any call-specific values. That is, the result set consists solely of those values defined in the abstract response class and "inherited" by the concrete [AddToltemDescription](#) response object.

Ending Items Early

The ideal conclusion to listing an item is to have that item sell. However, the seller may need to end an item listing before it would normally end. In the API, this is done using [EndItem](#).

Sellers can use **EndItem** to end any of their active eBay auctions. **EndItem** is used to terminate an auction prior to its ending normally based on:

- Auction's end time expiring
- Buyer exercising the Buy It Now option for an auction
- Purchase of a single-item, fixed-price auction
- Purchase of all of the items in multi-quantity, fixed-price listing
- Sale to the highest bidder

A seller might need to end the listing because the item is no longer available or was lost or broken.

Note: **EndItems** allows you to end up to ten (10) items early with a single call. Any restrictions that apply to **EndItem** apply to **EndItems**, as well. Refer to the [EndItems page](#) in the Call Reference for more information.

For auctions that have qualifying bids (i.e., current high bid that, when applicable, meets the minimum reserve price), you must specify an **EndingReason** value. If you specify "SellToHighBidder", the listing ends as a successful auction with winning bidder(s). If you specify another **EndingReason** value, such as "LostOrBroken", all bids are automatically cancelled before the item is ended as an unsuccessful auction. Starting October 1, 2012, sellers will be allowed to end one auction that have at least one bid early per calendar year with no fee. However, all other auctions ended early after the first occurrence will incur a fee. This fee will be equal to the Final Value Fee charged for the listing if the listing had ended naturally with the winning bid being the current high bid when the seller ended the listing prematurely.

If the listing has no bids (or no qualifying bids) when **EndItem** is used, then the listing is ended as an unsuccessful auction. When a listing is ended early and was not successfully sold, the seller must specify a reason for ending the listing early.

A single-quantity, fixed price listing can only be ended early if no user purchased that single item. Otherwise the listing would have ended on that purchase.

Multi-quantity, fixed-price listings present a distinct variation of the use of **EndItem** to end a listing early. The purchase of a single item in a fixed-price auction (where multiple items are offered) is immediate, but does not end the listing. **EndItem** can be called for such listings so long as the listing's end time has not passed. But any individual items in the listing that have already been purchased by buyers are still considered successfully sold when **EndItem** is used. Order line items cannot be undone by ending a listing early. Only the unpurchased items in the listing are ended when **EndItem** is used.

Note the following restrictions for ending items early:

- **EndItem** cannot be used to end another user's listings. The requesting user making the call must be the listing's seller. (Users are not allowed to end another user's listings—except, of course, by actually purchasing the item.)
- **EndItem** can be used only on listings that are still active. The call fails with an error if used with an already ended item listing.
- **EndItem** will fail with an error if there are 12 hours or fewer left before the auction ends and there are any bids on the auction, including a reserve met bid.

The table below summarizes the effects of ending an item early for various listing types.

Table 19-1 Effects of Ending an Item Listing Early

Listing Type	Effects
Auction	Current high bidder becomes winning bidder if the reserve price is met (if the listing has a reserve price) and EndReason is set to SellToHighBidder . The listing ends as a successfully sold item. If a different EndReason is specified, any bids are cancelled and the listing ends as an unsuccessful listing (item unsold).
Fixed-price	The listing ends. Any order line items are still valid and payment is still expected from the buyer(s) and shipping of the item(s) are still expected from the seller.
Buy It Now	Listing ends as an unsuccessful listing (item unsold). If a user had exercised the Buy It Now option, the listing would have already ended. If at least one user has placed a bid on the listing, treat as an auction (see above).
Classified Ad	The listing ends immediately, instead of at the scheduled date and time.

The application must identify the listing to be ended early. This is done by specifying that listing's item ID as the value for the **ItemID** property (which is of type **ItemIDType**) of the **object**. If using **EndItem** results in the listing ending as an unsuccessful listing (the item was not sold), then the application must specify the reason for ending the listing. This is done by specifying the reason in the **EndingReason** property of the request object. Possible values for the **EndingReason** property are enumerated in the **EndReasonCodeType** code list.

Working with Seller Events

To retrieve a list of a seller's items for which seller events have occurred, use [GetSellerEvents](#).

To specify a seller, use the eBay user ID for the seller. If the eBay user ID is not specified, seller events are retrieved for the seller identified by the authentication token.

You can use the following time-based filters to reduce the number of events returned: **ModTimeFrom**, **ModTimeTo**, **EndTimeFrom**, **EndTimeTo**, **StartTimeFrom**, and **StartTimeTo**. For example, to retrieve only

events for items that were modified during a specific time period, you can use **ModTimeFrom** and **ModTimeTo**. Best Practices are described in the following *Knowledge Base* article:

Best Practices for GetSellerEvents and GetSellerTransactions

https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=222

At least one of the following must be specified: **ModTimeFrom**, **EndTimeFrom**, or **StartTimeFrom**. Time-based filters can work in pairs (e.g., **ModTimeFrom** and **ModTimeTo**). If you do not specify the “To” filter, it is set to the time you make the call. For better results, the time period you use should be less than 48 hours.

The seller’s user ID and filtering are specified for the call using the following properties.

Table 19-2 Seller Event Data Specifiers

Item Property	Required	Type/Code List	Purpose
UserID	Y	UserIDType	eBay user ID for the seller whose events are to be returned.
StartTimeFrom	N/Y	dateTime	Describes the earliest (oldest) date to use in a date range filter based on item start time. Must be specified if StartTimeTo is specified.
StartTimeTo	N/Y	dateTime	Describes the latest (most recent) date to use in a date range filter based on item start time.
EndTimeFrom	N/Y	dateTime	Describes the earliest (oldest) date to use in a date range filter based on item end time. Must be specified if EndTimeTo is specified.
EndTimeTo	N/Y	dateTime	Describes the latest (most recent) date to use in a date range filter based on item end time.
ModTimeFrom	N/Y	dateTime	Describes the earliest (oldest) date to use in a date range filter based on the time an item’s record was modified. Must be specified if ModTimeTo is specified.
ModTimeTo	N/Y	dateTime	Describes the latest (most recent) date to use in a date range filter based on the time an item’s record was modified.
IncludeWatchCount	N	boolean	Specifies whether to include WatchCount in Item nodes returned. WatchCount is the number of watches buyers have placed on the item from their My eBay accounts.
NewItemFilter	N	boolean	Limits the returned events to just those where the modification to the item is its initial listing (i.e., only new item listings are returned).

The data returned by [GetSellerEvents](#) contains information about the items that fit your input filters. One [ItemType](#) object is returned for each item. See the schema documentation for [ItemArrayType](#) and [ItemType](#) for a complete list of their properties.

Table 19-3 GetSellerEventsCall Return Values

Returned Values	Type/Code List	Meaning
ItemArray	ItemArrayType	Collection of Item objects, each of which represents an item listing that incurred the type of seller event change specified in the call's inputs.

Your application traverses the returned array for the data in each [ItemType](#) object. You can compare each object to data previously returned. For example, an application could compare returned data with the data stored in a database table. If the item's current price had changed (such as for progressive bidding in a Chinese auction), your comparison would reveal the change.

Some baseline set of data is needed to determine what item values changed as this is not indicated in the property values of the returned [ItemType](#) objects. The [ItemType](#) objects hold only current values, as of the last change to the item. For some purposes, the data values returned are of value (such as seeing the current price of a competitive-bidding listing) without comparing the values to an older set of values. Many sellers wish to be able to make this determination to not only see what the current values are, but what the old values were before the seller event. To do that the application must store the returned values for a given item (such as to a database table) on each [GetSellerEvents](#). Subsequent executions of [GetSellerEvents](#) (using incrementally later date ranges) can then use this historical data in comparison with the newly returned data to highlight the changes.

Learning About Bidders

[GetAllBidders](#) (see “[GetAllBidders](#)” on page 801) enables you to:

- Identify all bidders on a listing, whether the listing has ended or not.

The [CallMode](#) value of [ViewAll](#) returns all bidders for an item. It is available for use by any user for whatever purpose they may have in mind.

- Identify all non-winning bidders for an ended listing.

The [CallMode](#) value of [EndedListing](#) returns all non-bidders for ended listings only. It is available for use by only sellers for whatever purpose they may have in mind.

- Identify all bidders on a listing who qualify for a second chance offer.

In the event that the winning bidder for a listing fails to complete the purchase of the item, the seller may wish to offer the item to one of the non-winning bidders for that listing. Or it might be that the seller has similar items to sell and wishes to offer those to other bidders on the first listing.

With some restrictions, a seller can extend a second chance offer to bidders in support of such scenarios. [GetAllBidders](#) plays a role in this. See “[Making Second Chance Offers for Items](#)” on page 426.

Putting Store Items on Sale

Sales are a great way to promote the items in your store. You can create sales that are seasonal, for specific holidays, for specific categories, or just to clear out inventory. Sales events promote your Store by creating excitement and encouraging buyers to browse and buy. Promotional sales apply to regular fixed-price format, auction, and auction/BIN formats. Auction and auction/BIN format listings can be added as free shipping sales only.

This section covers the following topics:

- “Promotional Sale API” on page 394
- “Defining a Promotional Sale” on page 394
- “Managing a Promotional Sale” on page 396
- “Promotional Sale Rules” on page 396
- “Markdown Manager” on page 397

Promotional Sale API

Using the promotional sale APIs, store owners can easily create price discount and/or free shipping sales for many listings at a time. The following API calls can be used to create and manage promotional sales:

- [SetPromotionalSale](#)
- [SetPromotionalSaleListings](#)
- [GetPromotionalSaleDetails](#)

Once you have established a promotional sale, several other calls, such as [GetItem](#), can retrieve the original price of a discounted item, along with the start-time and end-time of a discount. The data is returned in a [PromotionalSaleDetails](#) container. Additional information about retrieving this data is accessible from the following topics:

- “[GetItem](#)” on page 833
- “[GetSellerList](#)” on page 859
- “[GetBidderList](#)” on page 804
- “[GetMyeBayBuying](#)” on page 839
- “[GetMyeBaySelling](#)” on page 841

Defining a Promotional Sale

The first step in applying a discount across many listings is creating a promotional sale using [SetPromotionalSale](#). This call (see “[SetPromotionalSale](#)” on page 930) is used to add, modify, or remove a promotional sale. If you are adding a sale, the input fields include the action (add), sale name, sale type ([PromotionalSaleDetails.PromotionalSaleType](#)), and time period.

Sale Types

There are three types of promotional sales:

- Price discount only
- Free shipping only
- Price discount and free shipping

Regular fixed-price format listings can be added to any of the sale types. Auction and auction/BIN format listings can be added to free shipping sales only.

Price Discount

If your sale includes price discount, you will need to set the discount type as either a percentage reduction of the price or a fixed value to subtract from the price. Set [PromotionalSaleDetails.DiscountType](#) to "Percentage" for a percentage discount or "Price" for a fixed value price discount. For either type, you must specify the discount value ([PromotionalSaleDetails.DiscountValue](#)), which is the percentage price reduction for a percentage discount, or the specific amount taken off the price for a fixed price discount. For example, a value of "10" means a 10% price discount for a percentage discount or a \$10 discount for a fixed price discount. Fixed amount discounts will be in the currency of the original listing.

Free Shipping

When a listing is in a free shipping sale, the first shipping service cost will be set to free. Any other shipping services in addition to the standard free shipping will still be available. Only the first shipping service/cost appears at the top of the View Item page and in the Shipping column on Search Results. The other options would only show at the very bottom of the View Item page. This is consistent with the behavior for listings with standard free shipping.

Time Period

A start time ([PromotionalSaleDetails.PromotionalSaleStartTime](#)) and end time ([PromotionalSaleDetails.PromotionalSaleEndTime](#)) must be specified for most sales. You can schedule a Sale up to 6 months in advance, or you can schedule a sale to start as early as the next hour. Sale durations must be a minimum of 1 day and cannot exceed 45 days.

Sale end time does not apply to free shipping sales for auction-style listings. When free shipping is added to an auction-style listing it is required to stay on for the duration of the listing. The end date of the sale is synonymous with the end date of the listing, therefore entering a sale end date is not applicable.

Promotional Sale ID

When you successfully define a new promotional sale, the [SetPromotionalSale](#) call output includes the unique ID for the sale ([PromotionalSaleID](#)). The [PromotionalSaleID](#) is used as input for the following calls:

- [GetPromotionalSaleDetails](#) to get information about a sale
- [SetPromotionalSale](#) to modify a sale
- [SetPromotionalSaleListings](#) to add or remove listings for a sale

Managing a Promotional Sale

Once you have defined a promotional sale, you can (with some restrictions) add listings, change sale types, change sale start and end times, change price discounts, remove listings from the sale, and delete the sale.

Getting Information about Promotional Sales

You can get information about an existing promotional sale using [GetPromotionalSaleDetails](#) (see “[GetPromotionalSaleDetails](#)” on page 854). If you specify a [PromotionalSaleID](#), you get details for the specified promotional sale only, otherwise you get details for all of your promotional sales.

Adding or Removing Listings for a Promotional Sale

After you have added a promotional sale, use [SetPromotionalSaleListings](#) to add or remove listings for the sale. To add or remove listings, you must specify the sale ID ([PromotionalSaleID](#)) and set [Action](#) to “Add” or “Delete.”

When adding listings to a sale, you can specify an array of individual item listings ([PromotionalSaleItemIDArray.ItemID](#)) and/or you can specify criteria for including multiple listings, such as all fixed price items ([AllFixedPriceItems](#)) or all items listed in a specific store category ([StoreCategoryID](#)). When removing listings from a sale, you must specify an array of individual item listings ([PromotionalSaleItemIDArray.ItemID](#)) you want to remove.

See “[SetPromotionalSaleListings](#)” on page 931) for additional information.

Modifying a Sale

Use [SetPromotionalSale](#) to modify a sale. To modify a sale, you must specify the sale ID ([PromotionalSaleDetails.PromotionalSaleID](#)) and set [Action](#) to “Update.” Modification of a sale can include changing the sale type, changing the sale start and end times, and changing price discounts.

Deleting a Sale

Use [SetPromotionalSale](#) to delete a sale. To delete a sale, you must specify the sale ID ([PromotionalSaleDetails.PromotionalSaleID](#)) and set [Action](#) to “Delete.” You can delete a sale as long as it is not active or pending. Sales cannot be deleted if they contain auction-style listings that have bids.

Promotional Sale Rules

Refer to [Markdown Manager rules](#) in the eBay site help for rules and restrictions for promotional sales. See the [Markdown Manager](#) and [Creating a Free Shipping sale with Markdown Manager](#) sections of the eBay Stores FAQ page for additional information about the rules, restrictions, and behavior for promotional sales.

Refer to “[Promotional Sale Differences](#)” on page 610 for a list of promotional sale restriction and logic differences by eBay site.

Markdown Manager

The promotional sale APIs provide the same functions as Markdown Manager in the eBay Stores Toolkit. For more information about Markdown Manager, the benefits of promotional sales, and marketing tips, please see the Markdown Manager page in the eBay Stores advanced selling pages:

<http://pages.ebay.com/storefronts/markdownmanager.html>

For guidelines on using Markdown Manager, see Putting Store Items on Sale Using Markdown Manager in the eBay Stores help:

http://pages.ebay.com/help/specialtysites/items_on_sale.html

The eBay Stores FAQ has answers to common questions about Markdown Manager and sales:

http://pages.ebay.com/storefronts/faq.html#markdown_manager

Using Selling Manager Calls in the Trading API

Selling Manager and Selling Manager Pro are tools with advanced functionality for managing listings. The Trading API includes 26 calls that expose these tools' functionality. In the Trading API, each Selling Manager call has "SellingManager" as part of the name of the call.

The Selling Manager calls are subject to change without notice; the deprecation process is inapplicable to these calls.

If you write an embedded application, then your application is **integrated** directly into the eBay.com interface. But your use (in the **Trading API**) of the Selling Manager calls does **not** require that your application be embedded into the eBay.com interface.

For information about writing an embedded application (i.e., an application that is integrated directly into the eBay.com interface), see the following location:

<http://developer.ebay.com/products/selling-manager-applications/>

Subscribing a User to Selling Manager

Selling Manager calls can be made on behalf of a user, as long as the user has a subscription to Selling Manager (free) or to Selling Manager Pro. For many calls, a subscription to Selling Manager Pro is required. For information about specific Selling Manager calls, including subscription requirements, see the Trading API Call Reference:

Trading API Call Reference

<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/index.html>

For the information provided to end users about Selling Manager and Selling Manager Pro, see the following locations.

Selling Manager
<http://pages.ebay.com/selling%5Fmanager>

Selling Manager Pro
http://pages.ebay.com/selling_manager_pro

Concepts for Selling Manager Calls

You can use Selling Manager calls to help sellers manage listings based on sellers' folders, products, templates, and automation rules.

For the input and output fields of all the calls (including calls for revise and delete operations), see the Trading API Call Reference, accessible from [“Selling Manager Documentation”](#) on page 400.

Here is an overview of the concepts used in many of the calls:

- **Folder.** A container for similar products.
- **Inventory.** The products available for listing, along with their quantities, templates, and folders.
- **Product.** A type of merchandise. A product name is the same as a sale template group name. A product ID is the same as a sale template group ID. A Selling Manager product is listed using a Selling Manager template. The concept of “product” in Selling Manager calls is unrelated to the concept of “product” in non-Selling Manager calls (such as [FindProducts](#) in the Shopping API).
- **Template.** Contains the data needed to list an item. In Selling Manager Pro, a “template” can be thought of as a “listing template.” A template ID is accessible by calling [GetSellingManagerInventory](#).
- **Automation Rule.** A rule for automatically listing new items from a template, or for relisting existing items.

Retrieving Selling Manager Inventory and Templates

You can retrieve a seller's existing folders, inventory, and templates before presenting the seller with custom options for managing their listings.

Retrieving Inventory Folders

You can use the [GetSellingManagerInventoryFolder](#) call to retrieve a seller's current folder hierarchy. Your application should store the folder hierarchy, and the [FolderID-FolderName](#) pairs, in order to present folder-related options to the seller.

Retrieving Inventory

You can use the [GetSellingManagerInventory](#) call to retrieve a seller's current product, template, and inventory data. Your application needs this data to present inventory-related options to the seller.

Retrieving Templates

After obtaining at least one `SaleTemplateID`, e.g. using [GetSellingManagerInventory](#), you can use the [GetSellingManagerTemplates](#) call to retrieve the data for one or more of a seller's templates. This data enables your application to present template-related options to a seller.

Adding, Revising, and Deleting Products and Templates

Your application can perform add, revise, and delete operations on products and templates. For a list of the calls available for these operations, see the Trading API Call Reference link in “[Selling Manager Documentation](#)” on page 400.

Adding a Product

If a seller wants a new type of merchandise, i.e. a new product, use [AddSellingManagerProduct](#).

Adding a Template

Use [AddSellingManagerTemplate](#) when a seller wants to create a new set of listing data for a new kind of listing. When you use [AddSellingManagerTemplate](#), specify the product for which the template will be used by specifying a `ProductID`.

Using Selling Manager Automation Rules

A seller can use automation rules to automatically list and relist items that are based on a specific template. For example, a seller can specify that a set of items is automatically relisted unless the product inventory is at or below a certain level.

Retrieving Current Automation Rules

If you want to retrieve a template's automation rules, use [GetSellingManagerTemplateAutomationRule](#), specifying the template ID in the `SaleTemplateID` field.

Adding New Automation Rules

Using [SetSellingManagerTemplateAutomationRule](#), you can add an automation rule to an existing template. After you retrieve the seller's templates using the [GetSellingManagerTemplates](#) call, you specify the template ID (in `SaleTemplateID`) of the template to which you want to associate a rule. The rule is added to existing rules. To delete a rule, use [DeleteSellingManagerTemplateAutomationRule](#).

See the input fields for [SetSellingManagerTemplateAutomationRule](#) (accessible from “[Selling Manager Documentation](#)” on page 400) to see the available rules and the available settings for each rule.

Retrieving Selling Manager Alerts and Email Logs

A Selling Manager Application can provide a seller with alerts and email logs.

Retrieving Alerts

Use [GetSellingManagerAlerts](#) to retrieve up-to-date data about listings. The types of data you can retrieve are:

- **Automation alerts.** Alerts related to automation rules, e.g. that an automation rule that normally would have listed an item has failed to list the item.
- **General alerts.** General alerts, e.g. that negative feedback was received.
- **Inventory alerts.** Alerts related to inventory, e.g. that a product is low in stock.
- **PaisaPay alerts.** Alerts related to items paid with the PaisaPay payment method.
- **Sold alerts.** Alerts related to sold items, e.g. that payment for an item is overdue.

Retrieving Email Logs

Use [GetSellingManagerEmailLog](#) to retrieve a log of emails that a seller sent, or is scheduled to send, to a buyer. The log includes emails regarding the buyer's payment and feedback. For each email, the log indicates the state of the email (successful, pending, or failed).

Retrieving and Modifying Data About Sold Items

When an item is sold, Selling Manager generates a sale record. A sale record is displayed in the Sold view in Selling Manager.

The sale record has a sale record ID (also referred to as a sale record number) for each order line item. In the following calls, the value for the sale record ID is in the [SellingManagerSalesRecordNumber](#) field: [GetOrders](#), [GetOrderTransactions](#), [GetItemTransactions](#), and [GetSellerTransactions](#).

In the Selling Manager calls, the value for the sale record ID is in the [SaleRecordID](#) field.

Retrieving the Data for the Items Sold

Using a time-range or other filter in [GetSellingManagerSoldListings](#), you can get a list of a seller's sold items. This list includes comprehensive information about each item sold.

Retrieving and Revising a Record of Sale

You can use [GetSellingManagerSaleRecord](#) to retrieve a detailed, individual sale record for an item that was sold. You can use [ReviseSellingManagerSaleRecord](#) to revise many fields of a record, e.g. the pay date, sales tax, and shipping address and date.

Selling Manager Documentation

The 26 Selling Manager calls provide functionality that is provided by Selling Manager and Selling Manager Pro.

Links to the input and output of the Selling Manager calls are available in the Table of Contents of the Trading API Call Reference:

Trading API Call Reference

<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/index.html>

Chapter 20

Communication Between Members

This section contains the following subsections:

- “[Summary of Messaging Calls](#)” on page 402
- “[Static Email Addresses in Trading API Calls](#)” on page 402
- “[Sending Messages](#)” on page 403
- “[Retrieving Messages](#)” on page 404
- “[Example: Responding to a Question](#)” on page 404
- “[Enabling Communication Through Skype](#)” on page 405

Summary of Messaging Calls

The following calls are used in communication between members.

AddMemberMessageAAQToPartner: Enables a buyer and seller on either side of an order to send messages to each other within 90 days of the creation of the order. Messages are sent to the eBay user’s My Messages InBox.

AddMemberMessageRTQ: Enables a seller to reply to a question about an active item listing. This message is sent to the eBay user’s My Messages InBox.

AddMemberMessagesAAQToBidder: Enables a seller to contact both bidders and users who have made offers (via Best Offer) during an active listing. This message is sent to the eBay user’s My Messages InBox.

GetMemberMessages: Retrieves messages posted to the Ask Seller A Question messaging system. Buyers can ask sellers questions via the website, and sellers can respond to these ASQ messages by using this call, and the related **AddMemberMessageRTQ** call, to send them email.

GetMyMessages: Retrieves the given user’s messages. The information returned can vary depending on the detail level requested.

Static Email Addresses in Trading API Calls

Each eBay member is assigned a static alias to be used in a static email address. The email address ends in `@members.ebay.<SiteDomain>`. The `<SiteDomain>` value is based on the user’s registered site, e.g. `@members.ebay.de`. The `<SiteDomain>` value identifies the user’s language. If the user changes the registered site, then the value of `<SiteDomain>` changes accordingly.

The static alias is used in the “reply to” address in emails from other members. The email address returned by **GetMemberMessages** is a static email address, in the following field:

MemberMessage.MemberMessageExchange.Question.SenderEmail. Additionally, the conversation ID is in the body of the email returned by **GetMemberMessages**. The conversation ID also is referred to as the email reference ID. All messages must contain a conversation ID, so eBay can validate them.

In the following calls, the current email address that is returned is supplemented by a static email address, as follows:

- **GetSellerTransactions.Transaction.Buyer.Email** is supplemented by **TransactionArray.Transaction.Buyer.StaticAlias**.
- **GetItemTransactions.Transaction.Buyer.Email** is supplemented by **TransactionArray.Transaction.Buyer.StaticAlias**.
- **GetOrderTransactions.Order.TransactionArray.Transaction.Buyer.Email** is supplemented by **OrderArray.Order.TransactionArray.Transaction.Buyer.StaticAlias**.
- **GetMyeBaySelling.Transaction.Buyer.Email** is supplemented by **OrderArray.Order.TransactionArray.Transaction.Buyer.StaticAlias**.

Sending Messages

Use of the API to send a message from one eBay member to another depends on the relationship between the members.

[“Messages Sent Regardless of Relationship” on page 403](#)

[“Messages Sent When an Order Relationship Exists” on page 404](#)

[“If the Users Have a Bidding Relationship” on page 404](#)

Messages Sent Regardless of Relationship

All eBay users can send messages to other eBay users through the eBay Web site. The messages appear in the My Messages InBox. See the following in eBay’s online help:

Contacting eBay Members

<http://pages.ebay.com/help/tp/contacting-members.html>

How Sellers Answer Questions

http://pages.ebay.com/help/sell/answer_qs.html

If a seller has received a question about an active listing (from someone sending a message via the web site), the seller can use **AddMemberMessageRTQ** to reply to that question. No order or bidding relationship is required. The seller has the option of posting the question and response in the listing for all prospective buyers to see. See “[AddMemberMessageRTQ](#)” on page 785. See also “[Example: Responding to a Question](#)” on page 404.

Messages Sent When an Order Relationship Exists

An order relationship exists between two members when one is the seller of an item and the other is a buyer (winning bidder in an auction). A single or multiple line item order is created once there is a commitment to purchase from the buyer, regardless of whether payment has been made.

[AddMemberMessageAAQToPartner](#) enables a buyer and seller in an order relationship to send messages to each other within 90 days of the creation of the order. See “[AddMemberMessageAAQToPartner](#)” on page 784. (AAQ stands for Ask A Question.)

If the Users Have a Bidding Relationship

[AddMemberMessagesAAQToBidder](#) enables a seller to contact a bidder who bid on or a user who made a Best Offer on an active listing. Such a message is sent to the eBay user’s My Messages InBox. See “[AddMemberMessagesAAQToBidder](#)” on page 786. (AAQ stands for Ask A Question.)

If the item is in the Motors or Business & Industrial categories, and if one member is the seller and the other has made a bid on the seller’s item, [GetUserContactDetails](#) can be called for either user to obtain the other’s phone number, if available. See “ [GetUserContactDetails](#)” on page 877.

Retrieving Messages

Any user can retrieve his messages via My eBay on the eBay Web site.

Messages can also be retrieved for any user via [GetMyMessages](#) or [GetMemberMessages](#). See “[GetMyMessages](#)” on page 842 and “[GetMemberMessages](#)” on page 837.

eBay users can subscribe to the [MyMessages](#) platform notification in order to be automatically notified when specific kinds of messages are received in their My Messages InBox. See “[MyMessages Notification](#)” on page 737.

Example: Responding to a Question

Here is a typical approach for enabling a seller to respond to questions from prospective bidders:

- 1 Your application lists an item for sale. The [ItemID](#) is returned from the listing.
- 2 A buyer uses the eBay Web site to ask the seller a question about the listed item. The [MessageType](#) for this message is internally set to [AskSellerQuestion](#).
- 3 eBay stores the message, marking it as Unread.

Your application checks [Item.ListingDetails.HasUnansweredQuestions](#) and [Item.ListingDetails.HasPublicMessages](#) for the [ItemID](#).

- 4 Your application calls [GetMemberMessages](#) to retrieve the [MessageIDs](#) of any unanswered messages for the [ItemID](#).

- 5 If there are messages, your application retrieves the **MessageID** so it can respond to the buyer by posting a message to the website.
- 6 Your application calls **AddMemberMessageRTQ**, providing the option to the seller of whether to post the question and the seller's response for all to see, and whether to send a copy of the email to the seller's personal email address.

Enabling Communication Through Skype

For listings in some categories on some sites, you can add Skype contact information. You also can retrieve Skype names linked to an eBay account and you can retrieve a Skype name that is part of a listing.

Skype enables voice and other communication over the Internet. If you were using Skype functionality before Schema version 465, please see “[Changes in Skype Functionality](#)” on page 406.

The prerequisites for adding Skype buttons to listings (see “[Making Skype Available In Listings](#)” on page 405) are:

- Skype is supported under all the categories of the listing, and on the site on which you are adding the listing. To determine if these prerequisites are satisfied, use the **SkypeMeTransactionalEnabled** (for Classified Ads) and/or the **SkypeMeNonTransactionalEnabled** (for Sale listings) flags as filters in the **GetCategoryFeatures** call (see “[GetCategoryFeatures](#)” on page 808).
- The seller has a Skype account and has linked it (on the eBay site) with his or her eBay account.

You can utilize the **User.SkypeID** field of **GetUser** (see “ [GetUser](#)” on page 876) to retrieve up to three Skype names linked to an eBay account. If you have added Skype contact information to a listing, **GetItem** (see “ [GetItem](#)” on page 833) and **GetSellerList** (see “ [GetSellerList](#)” on page 859) return the information as part of the data in the **Item** object.

General information about Skype is available here:

Skype
<http://www.Skype.com>

Making Skype Available In Listings

If the prerequisites for adding Skype buttons to listings are met (see “[Enabling Communication Through Skype](#)” on page 405), you can make communication through Skype available in listings.

For example, in **AddItem**, you can include the Skype-related fields of **ItemType**. Specifically, you can include **SkypeEnabled** and **SkypeID**, and include contact options such as **Chat** and **Voice**, as in the following example.

Example 20-1 Sample XML from AddItem for Including a SkypeMe Button in a Listing

```
...
<Item>
...
<SkypeEnabled>true</SkypeEnabled>
<SkypeID>MySkypeName</SkypeID>
<SkypeContactOption>Chat</SkypeContactOption>
<SkypeContactOption>Voice</SkypeContactOption>
...
</Item>
...
```

Changes in Skype Functionality

Before Schema version 465 (see “[Standard Data for All Calls](#)” on page 92), Skype functionality allowed the addition of Skype buttons to listings on several international sites. In **AddItem**, before Schema version 465, **SkypeOption** in **ItemType** indicated Skype contact options. The following field is now deprecated and it has been deleted from the latest schema: **SkypeOption** in **ItemType**.

Now, **SkypeContactOption** is used instead of **SkypeOption**. For the code that is used to add Skype buttons beginning with Schema version 465, see “[Making Skype Available In Listings](#)” on page 405.

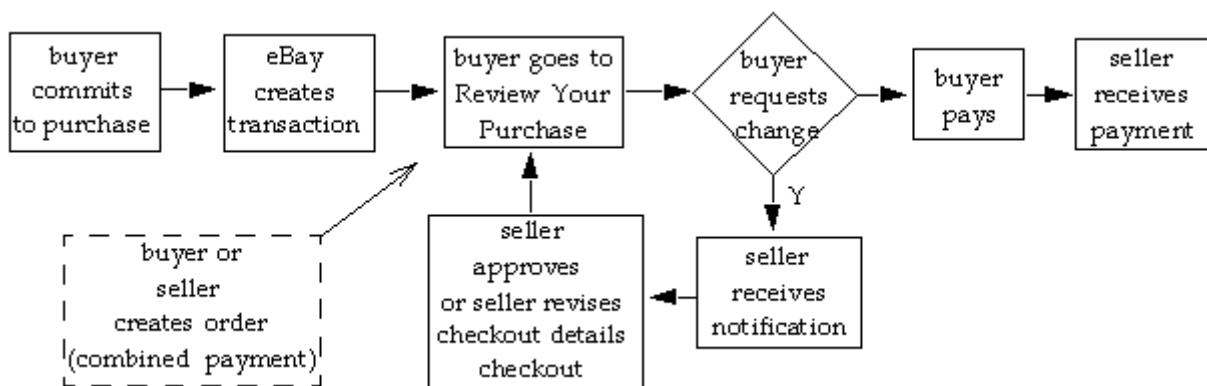
Chapter 21

Checkout

The eBay checkout flow involves order line items, which are the basic building block of the eBay system (see “[Order Line Items](#)” on page 145). The **Transaction** object contains information about the sale of one order line item by one buyer from a single auction or fixed-price listing. In the case of fixed-price listings, a buyer can purchase multiple, non-variation items, but these multiple items are still considered one order line item.

An eBay order line item can be considered a contract between two eBay users: a buyer and a seller. Once a buyer makes a commitment to buy an item, it is the buyer’s responsibility to pay for the order, and it is the seller’s responsibility to ship the item (as it is described in the listing) to the seller.

The eBay checkout flow is captured in the following diagram:



This chapter discusses the eBay checkout flow and the different states of orders. It discusses how sellers and applications interact with and modify order data from the creation of an order line item to the shipment of the item.

[“Creation of an Order Line Item” on page 407](#)

[“Listing Fields and Features Affecting an Order Line Item” on page 408](#)

[“Reviewing and Modifying an Order Line Item” on page 411](#)

[“Buyer Payment” on page 414](#)

Creation of an Order Line Item

An order line item is created in the eBay system after the buyer has committed to purchase a seller’s item. The timing and logic of the buyer’s “commitment to purchase” depends on the listing type, as described in the table below.

Table 21-1 Commitment to Purchase eBay Items

Listing Type or Feature	Order Line Item Creation Time
Auction listing	An order line item is created at the end of the auction between the seller and the highest bidder. If a Reserve Price is set on a listing and is not met by the highest bidder, the listing ends without a winning buyer.
Fixed-price listing	An order line item is created once the buyer clicks the Buy It Now button on the listing page. Only one order line item is created even if the buyer is buying multiple, non-variation fixed-price items.
Buy It Now in Auction Listing	An order line item is created once the buyer clicks the Buy It Now button on the listing page. Note that the Buy It Now option is no longer available once a bid is made on an auction listing. If a buyer uses the Buy It Now option on an auction listing, the auction automatically ends regardless of how much time is left.
Best Offer	A order line item is created when the seller accepts the buyer's Best Offer or Counter Offer price, or if the buyer accepts the seller's Counter Offer price.
Second Chance Offer	An order line item is created when a previous bidder (on the seller's listing) decides to accept a seller's Second Chance offer and clicks the Buy It Now button. Second Chance offers can appear on a closed listing page, in My eBay Messages , or under the Didn't Win tab of My eBay .

For auction listings, the winning bidder receives a confirmation email right after the auction ends. The email includes a link to the **Pay now** page.

Note: eBay processes some order line items with backend batch jobs, which can create a small discrepancy between the time stamp of the listing event that spawns the order line item and the time stamp of the initial order line item entry in the eBay database.

Listing Fields and Features Affecting an Order Line Item

When a seller lists an item through the eBay site or through the API, there are numerous fields/features that ultimately affect the order line item between the buyer and seller. The API fields and their descriptions are listed in the table below.

Table 21-2 Listing Fields Affecting a Transaction

Listing API Field	Purpose
Item.AutoPay	Flag to turn on the immediate payment requirement. For this feature to be available, the item's category must support immediate payment and the payment method used must be PayPal. See " "Requiring Immediate Payment" on page 187 for more information and requirements on this feature.
Item.BestOfferDetails.BestOfferEnabled	Flag to turn on the Best Offer feature. This feature is only applicable for fixed-price listings. See " "Enabling Best Offer" on page 225 for more information on this feature.
Item.BuyItNowPrice	This field sets the Buy It Now price in an auction listing. The Buy It Now feature is optional for a seller and becomes unavailable as soon as the first bid is made and the bid is higher than the Reserve Price (if one is set by the seller)..
Item.DispatchTimeMax	This field specifies the maximum number of business days a seller has to prepare an item for shipment once payment has been made for the order. This time does not include shipping time. This field is required for most categories.

Table 21-2 Listing Fields Affecting a Transaction

Listing API Field	Purpose
Item.GetItFast	Flag to enable Get It Fast shipping rules for a fixed-priced listing or for an auction item that is purchased with the Buy It Now feature. For the listing to be qualified for the Get It Fast feature, DispatchTimeMax must be set to '1' (or '0' once Same Day Handling becomes available) and the seller must offer at least one domestic expedited shipping service.
Item.ListingDetails.BestOfferAutoAcceptPrice	This field sets the minimum threshold value at which a buyer's Best Offer price will be automatically accepted by the seller. See " Automatically Declining or Accepting an Offer " on page 227 for more information.
Item.ListingDetails.MinimumBestOfferPrice	This field sets the minimum Best Offer price that a seller is willing to accept from a buyer. Any offer below this value is automatically rejected by the seller. See " Automatically Declining or Accepting an Offer " on page 227 for more information.
Item.ListingType	The default listing type is an auction (Chinese is the enumeration value). This field is not required for sellers wishing to create an auction listing. For all other listing types, like fixed-price (FixedPriceItem) or Half.com (Half), this field must be included in the call.
Item.PaymentMethods	This field is used to specify one or more accepted payment methods. Every listing must have at least one valid payment method. See " Payment Methods " on page 179 for more information.
Item.PayPalEmailAddress	A valid PayPal email address must be specified in this field if PayPal is one of the specified payment methods. This must be the email address that is linked to the seller's PayPal account.
Item.ReservePrice	This field sets the Reserve Price, or the lowest price the seller is willing to accept for an item in an auction listing. If the highest bid does not equal or exceed the Reserve Price at the end of the auction, the auction ends without a winning bidder. The Reserve Price feature is optional for a seller.
Item.ReturnPolicy	This container sets the Return Policy for an item. See " Offering a Clear Return Policy " on page 223 for more information on creating a Return Policy for a listing.
Item.ShippingDetails.ShippingType	This field indicates the shipping cost model the seller has set for the item. See " Specifying Shipping Types and Costs " on page 336 for more information on specifying shipping services and options when listing an item.
Item.ShippingDetails.CalculatedShippingRate	This container is required if the seller is offering a shipping service that uses calculated shipping to determine the final shipping costs. See " Specifying Calculated Shipping " on page 341 for more information on calculated shipping.
Item.ShippingDetails.ExcludeShipToLocation	This field allows sellers to specify one or more regions, countries, or domestic locations (for example, Alaska/Hawaii) as places they will not ship to. Additionally, US sellers can block PO Boxes (by specifying "PO Box" in this field) and German sellers can block DHL Packstations (by specifying "Packstation" in this field). If the prospective buyer's ship-to location is on this "exclusion" list, the buyer is returned an error if they attempt to buy or place a bid on the item. The value(s) specified in this field overrides the excluded ship-to locations specified in the seller's My eBay's Shipping Preferences. Sellers should see " Specifying Locations to Where You Ship " on page 324 for more information on controlling where they will ship an item.
Item.ShippingDetails.InsuranceDetails	This container is used to specify domestic shipping insurance costs and is only applicable for the AU, FR, and IT sites. See " Shipping Insurance " on page 362 for more information.
Item.ShippingDetails.InternationalInsuranceDetails	This container is used to specify International shipping insurance costs and is only applicable for the AU, FR, and IT sites. See " Shipping Insurance " on page 362 for more information.

Table 21-2 Listing Fields Affecting a Transaction

Listing API Field	Purpose
Item.ShippingDetails.PromotionalShippingDiscount	Flag to allow buyers to use promotional shipping discounts. See “ Shipping Cost Discount Profiles ” on page 356 for more information on setting and managing shipping cost discounts.
Item.ShippingDetails.InternationalPromotionalShippingDiscount	Flag to allow International buyers to use promotional shipping discounts. See “ Shipping Cost Discount Profiles ” on page 356 for more information on setting and managing shipping cost discounts.
Item.ShippingDetails.ShippingDiscountProfileID	This field is used by the seller to specify the shipping discount profile to use for the listing. See “ Shipping Cost Discount Profiles ” on page 356 for more information on setting and managing shipping cost discounts.
Item.ShippingDetails.InternationalShippingDiscountProfileID	This field is used by the seller to specify the International shipping discount profile to use for the listing. See “ Shipping Cost Discount Profiles ” on page 356 for more information on setting and managing shipping cost discounts.
Item.ShippingDetails.InternationalShippingServiceOption	This field is used by the seller to specify one to five International shipping service options the buyer may select. See http://pages.ebay.com/help/pay/ups-services.html for more information on domestic and International shipping services.
Item.ShippingDetails.ShippingServiceOptions	This field is used by the seller to specify one to four shipping service options the buyer may select. If one or more InternationalShippingServiceOption fields are used, one or more ShippingServiceOptions fields are required. See http://pages.ebay.com/help/pay/ups-services.html for more information on domestic and International shipping services.
Item.ShippingDetails.PaymentInstructions	This field is used by the seller to post additional payment instructions to the buyer. These instructions will appear on the View Item and checkout pages.
Item.ShippingDetails.RateTableDetails	This container specifies the shipping rate tables that are to be applied to a listing. Sellers can configure rate tables in My eBay shipping preferences to tailor the flat shipping rates offered for an item to be appropriate to each individual domestic region. To apply a domestic rate table, the value of the DomesticRateTable field must be set to Default . For more information, see “ Using Shipping Rate Tables ” on page 338.
Item.ShippingDetails.SalesTax	The seller can optionally use this container to specify the sales tax percentage that will be added to buyer’s cost for the item. One or more tax jurisdictions may be specified. This container is only applicable on the US and eBay Motors sites. If a sales tax table has already been established in My eBay Payment Preferences or through the SetTaxTable call, the seller can omit this field and just set the Item.UseTaxTable field to true. See “ Enabling Multi-jurisdiction Sales Tax ” on page 231 for more information on setting up and using sales tax tables.
Item.ShipToLocations	This field allows sellers to specify the regions and countries to which they will ship an item. Sellers can use this field to specify that they will ship to a region (for example, Asia), and then also use the Item.ShippingDetails.ExcludeShipToLocation field to exclude a country (for example, Afghanistan) within that region (Asia).. See “ ShipToLocation(s) ” on page 325 for more information about using this field.
Item.StartPrice	This required field sets the Start Price for an auction listing, or the fixed price for a fixed-price listing.
Item.UseTaxTable	If a sales tax table has already been established in My eBay Payment Preferences or through the SetTaxTable call, this flag indicates that the sales tax table will be used to determine the sales tax amount that should be applied to the item. If this field is used, the Item.ShippingDetails.SalesTax container should not be used. See “ Enabling Multi-jurisdiction Sales Tax ” on page 231 for more information on setting up and using sales tax tables.

Reviewing and Modifying an Order Line Item

After the order line item has been created between the buyer and seller, the checkout process involves the buyer reviewing the purchase, selecting the shipping and payment method (if multiple options are given by the seller), and payment. Before payment is made, the buyer also has the option of contacting the seller to ask a question about the item(s), or to request more information about payment or shipping.

Review Your Purchase Page

The **Review your purchase** page is the final page a buyer sees before proceeding to actually making the payment. On the **Review your purchase** page, the buyer can:

- Review the final cost of the order line item
- Request the total price of the order line item (including shipping and handling) from the seller
- Modify the shipping address
- Select a shipping service option (if multiple options are offered by the seller)
- Select one of the payment methods (if multiple options are offered by the seller)
- Request a change in final price (if allowed by the seller upon listing).
- Send a message to the seller
- Redeem a gift certificate or coupon

Buyer and Seller Communication Prior to Payment

The buyer may contact the seller prior to payment. The buyer may want to request more information on available payment or shipping methods, or possibly ask a question about the final price. The buyer can send a message to the seller through one of the checkout screens or through My Messages in My eBay. A buyer and seller with an order relationship can also communicate through the API using the [AddMemberMessageAAQToPartner](#) call. See “[Messages Sent When an Order Relationship Exists](#)” on page 404. The seller can approve any changes and/or provide more information to the buyer through the same communication channels.

Revising Checkout Details

The seller can use the Web site or [ReviseCheckoutStatus](#) to make the following checkout revisions:

- Supply additional details or instructions to the buyer
- Adjust the final price of the order line item
- Modify payment methods
- Modify shipping service options
- Modify the shipping address

The **Review your purchase** page is updated once the seller makes any changes to the final price, shipping method, or payment method.

[ReviseCheckoutStatus](#) is also used to revise the payment status of an order.

Creating a Combined Payment Order

A Combined Payment order can be created by a buyer or seller if two or more pending (not paid for) order line items exist between the same buyer and seller. Creating a Combined Payment order is a method of passing shipping discounts from the seller to the buyer, since it may be possible for the seller to ship multiple items in the same package, thus saving on shipping costs.

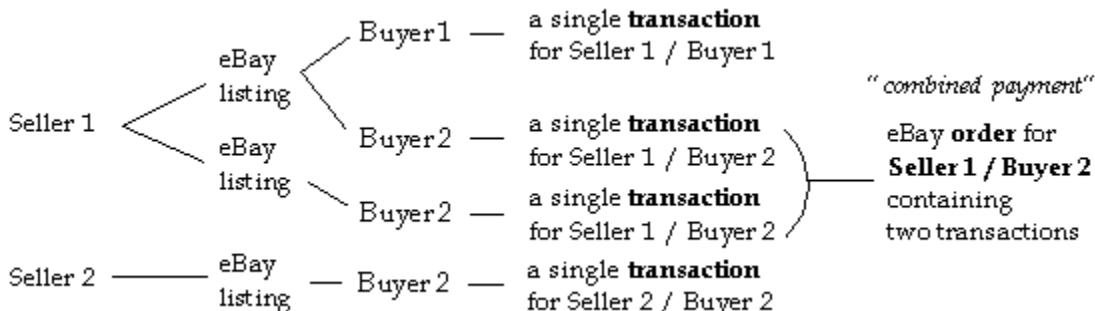
If multiple order line items are grouped into a Combined Payment order, the seller can send a single invoice to the buyer. See [SendInvoice](#). The buyer pays for the order in a single checkout or payment process.

An order can be created through the eBay Web site or with [AddOrder](#). For orders to be created by the buyer or seller, the Combined Payment option must be turned on for the seller's account. See ["Combined Payment"](#) on page 183 for more information.

Establishing Buyer-Seller Matches

Trading API calls can be used to match up buyers and sellers with multiple order line items with one another. To investigate the possibility of a buyer order, use [GetMyeBayBuying](#) and scan the **ItemArray** container to determine if two or more items won by the buyer involved the same seller. To investigate the possibility of a seller order, use [GetSellerTransactions](#) and scan the **TransactionArray** container to determine if two or more items sold by the seller were purchased by the same buyer.

The following figure graphically demonstrates the concept of buyer-seller matches. In the figure, there are two buyers and two sellers, and only *Seller 1* and *Buyer 2* have more than one order line item in common.



Determining Whether an Item Is Eligible to Include in an Order

An order line item must meet the following basic requirements before it can be combined with other similarly qualified order line items into a Combined Payment order:

- The buyer and seller are the same for all order line items.
- The same currency is used for all of the order line items.
- Order line items cannot be added to a Combined Payment order in certain states. See “[Order Status Values](#)” on page 413.
- None of the order line items are of these types:
 - eBay Motors vehicle (automotive parts and accessories are eligible)
 - Item specified by the seller as an Immediate Payment item
 - Real estate item
 - Ad-format item (special non-auction, real estate advertisement listings)

Order Status Values

When a Combined Payment order is created, it is assigned a status of [Active](#). The status of a Combined Payment order can be retrieved by using [GetOrders](#). The value is stored in the [OrderStatus](#) field. The Order Status values applicable to eBay checkout are discussed in the following table.

Table 21-3 Order Status Values

Order Status	Meaning
Active	The Combined Payment order is active but has not gone through PayPal or other payment process.
Inactive	A Combined Payment order becomes inactive if it is superseded by another Combined Payment order. Possible reasons for a Combined Payment order becoming inactive include the following: <ul style="list-style-type: none"> ■ The seller created a Combined Payment order including at least one order line item contained in a buyer-created Combined Payment order. ■ An existing buyer-created Combined Payment order becomes inactive if the same buyer creates a new Combined Payment order that contains at least one order line item contained in the existing Combined Payment order. ■ The buyer uncombined the order line items of an Active Combined Payment order.
Completed	The buyer has completed payment via PayPal or other payment method.
Cancelled	The Combined Payment order enters a cancelled state for the following reasons: <ul style="list-style-type: none"> ■ The Combined Payment order contains a order line item that is part of a Completed Combined Payment order ■ The Combined Payment order contains one or more order line items that are no longer eligible to be combined into a Combined Payment order

Buyer Payment

An integral part of the sale of an item is the buyer paying the seller for the purchase. How this is done depends in part on the payment methods the seller offers to the buyer (specified at listing time) and which one of these payment methods the buyer actually selects.

If the buyer selected PayPal as the method to pay for the item (assuming PayPal was one of the payment methods offered by the seller), the buyer is redirected from the **Review your purchase** page to the PayPal payment page. The buyer pays on the PayPal site. Then, PayPal transfers the funds for the buyer's payment to the seller's PayPal account. The seller's PayPal account is identified by the PayPal email address that the seller specified at listing time (or [Item.PayPalEmailAddress](#) in [AddItem](#)). For general information about PayPal, see "[PayPal](#)" on page 153. PayPal marks the order line item as completed once the buyer makes payment. That is, PayPal will set the payment status to completed (but see "[Holds on PayPal Payments](#)" on page 421).

Note: Starting in late August 2011, a new payment process is being implemented on the German (DE) and Austrian (AT) sites that will affect sellers new to listing on either of these two sites. In this new payment process, new sellers can only specify PayPal as the payment method, and buyers actually pay eBay and not the sellers for an order. After a holding period, eBay releases the funds to the sellers' accounts.

If a buyer selects a payment method other than PayPal, then arranging payment for the item is the seller's responsibility. If the payment method is an online payment service other than PayPal, then completing the payment may be a matter involving the seller, the buyer, and the payment service vendor. In these cases (when PayPal is not used), the order line item's status is marked as completed as soon as the buyer finishes the checkout process. Note that this is true whether or not payment has actually been received by the seller, since that is outside the ability of eBay to track.

Chapter 22

Completing the Sale

This section discusses orders and how to work with them during the processing that occurs after a winning bidder is determined (for auction listings) or a purchase is made (for fixed-price listings). This section also discusses the major activities a seller is concerned with upon the completion of a sale. These activities are partially or wholly supported by applications. Some aspects of some of these activities cannot, or are not, performed with an application and instead must be done on the eBay site.

Applications retrieve whole lists of order line items for a particular seller using [GetSellerTransactions](#). Retrieving the order line item data for a specific item is done with [GetItemTransactions](#). However, it is more efficient to use [GetOrders](#) to retrieve all single and multiple line item orders.

One reason an application returns a group of orders is to display the order line item data in a visual report. An end user can use this report to take various actions, such as contact some buyers to send them information.

Another reason for retrieving groups of orders is for processing the data outside the context of a user interface or human intervention. For example, an application could retrieve a list of the orders to assess activity, and only alert the seller if a predetermined condition exists (e.g., a buyer requests the total amount to pay, including shipping and insurance).

In addition, if your membership tier supports eBay Platform Notifications, and your application is certified for Production use (see [“Compatible Application Check”](#) on page 67), you can have eBay send certain kinds of order line item data to you immediately.

After an item is sold, the seller must ship that item to the buyer. The buyer and seller should leave each other feedback. Applications submit feedback with [LeaveFeedback](#) or [CompleteSale](#). To retrieve feedback for a user, use [GetFeedback](#).

- “Retrieving Order Line Item Data and Managing Orders” on page 416
- “Understanding Buyer Payment” on page 420
- “Holds on PayPal Payments” on page 421
- “Handling Non-Paying Bidders” on page 424
- “CompleteSale API Call” on page 424
- “Shipping the Item” on page 424
- “Final Value Fees and Credits” on page 425
- “Leaving and Getting Feedback” on page 426
- “Making Second Chance Offers for Items” on page 426

Retrieving Order Line Item Data and Managing Orders

The Trading API offers four calls that let you retrieve order line item data directly. The four calls are summarized below:

- **GetOrders** – this call returns all orders in which the authenticated caller is a buyer or seller. The types of orders returned include single line item orders, multiple line item orders, and Half.com orders. The caller may filter by OrderID, order role, or order status. The caller can also restrict the results to Half.com orders. Finally, there are time filters to restrict results based on order creation time (date range and number of days since creation) or the last modification date.
- **GetSellerTransactions** – this call returns all order line items for the seller making the request. The seller can filter results by seller-defined SKU (stock keeping unit) values. The caller can also restrict the results with time filters, including order creation time (number of days since creation) or the last modification date.
- **GetItemTransactions** – this call can be used in a variety of ways. It can return all order line items based on a particular ItemID, return the data for a particular order line item, or can return the order line item data for all variations of a multi-variation fixed-price listing. The caller can also restrict the results with time filters, including order creation time (number of days since creation) or the last modification date.
- **GetOrderTransactions** – this call returns all orders in which the authenticated caller is a buyer or seller. The caller may filter by OrderID, SKU value, or by OrderLineItemID. This call does not have any time filters

In addition, a qualifying application can use an optional feature called *Platform Notifications*, wherein eBay pushes order line item data to a URL that the application monitors. See “[Subscribing to Platform Notifications](#)” on page 417.

Retrieving Order Line Item Data Directly

GetOrders retrieves all orders in which the authenticated caller is involved as a buyer or seller. The retrieved orders can include single line item orders, multiple line item orders, and Half.com single and multiple line item orders. Based on the buyer or seller’s activity, the response can be quite large. To restrict the number of returned results, the caller can use the following types of filters:

- **CreateTimeFrom/CreateTimeTo** – use this filter set to filter by order creation date.
- **ModTimeFrom/ModTimeTo** – use this filter set to filter by the last modified date of the order.
- **Status** – use this filter set to filter by the status of the order. For example, you can restrict results to Active (unpaid) or Completed (paid) orders.
- **ListingType** – include and set this filter to **Half** to restrict results to Half.com orders only.
- **OrderID** – specify one or more **OrderID** values to return data on specific orders.

Alternatively, **GetItemTransactions** can be used to retrieve all order line items for a specific listing based on an **ItemID** (see “[Retrieving the Order Line Items for a Single Item](#)” on page 418). A caller

should also use [GetItemTransactions](#) when the application's user needs to retrieve order line items for an item that was listed by someone else (another seller). [GetOrders](#) and [GetSellerTransactions](#) only retrieve data for the caller making the request.

Best Practice for GetOrders

In the past, [GetOrders](#) was primarily used to retrieved Combined Payment and Half.com orders. Now [GetOrders](#) can be used to retrieve single line item orders (traditionally known as "transactions") and multiple line item orders (traditionally known as Combined Payment orders). Due to this new capability in [GetOrders](#), it is no longer essential to call [GetSellerTransactions](#) (to retrieve any Combined Payment order IDs) before calling [GetOrders](#).

To check the status of and process all orders on a continuous basis, make a call to [GetOrders](#) every 15 minutes. In the request, make sure:

- The [DetailLevel](#) is set to [ReturnAll](#)
- The [ModTimeFrom](#) value is set back two minutes from the time you last made the call
- The [ModTimeTo](#) value is set to the current time (if this field is omitted, the [ModTimeTo](#) value defaults to the current time)

Iterate through each order ([OrderArray.Order](#)) returned in the call and perform the following checks:

- 1 Check the [OrderArray.Order.CheckoutStatus.Status](#) value for each order. If the value is [Complete](#), then the order is ready for post-sale processing. If the value is [CheckoutIncomplete](#) or [Pending](#), [proceed to Step 2](#).
- 2 Check the [OrderArray.Order.CheckoutStatus.eBayPaymentStatus](#) for each order. If the value is [NoPaymentFailure](#), [PayPalPaymentInProcess](#), or [PaymentInProcess](#) (Germany only), the buyer has not paid yet or the payment is in pending mode. If any other value is returned (for example, [BuyerCreditCardFailed](#) or [BuyerECheckBounced](#)), there is an issue with buyer payment.
- 3 Check the other fields in the [CheckoutStatus](#) container to troubleshoot a buyer payment issue. For example, check the following:
 - If the value of the [CheckoutStatus.PaymentMethod](#) field matches a payment method specified in [OrderArray.Order.PaymentMethods](#).
 - If the value of the [CheckoutStatus.PaymentMethod](#) is [IntegratedMerchantCreditCard](#), the [CheckoutStatus.IntegratedMerchantCreditCardEnabled](#) field should be present and set to true.

If an order is not completely ready for post-sales processing, you need to allocate the inventory for the order line item(s), but you must wait to complete checkout. Keep in mind that it is also possible than some order information may change when checkout is still incomplete, such as when the buyer changes their shipping address. Item details, such as the [ItemID](#) and quantity purchased will not change, so you can manage your inventory while you wait to complete the sale and move on to fulfillment.

Subscribing to Platform Notifications

In addition to using the API calls, a qualifying application can have eBay send order line item data to a URL that the application monitors. This service is provided through eBay Platform Notifications.

eBay Platform Notifications is an optional feature that alerts qualifying applications when certain events occur on the Production eBay site. Platform notifications are triggered by events that occur when an order line item has been created, feedback has been left, or certain other changes have occurred on an order line item. Your application can use the data you receive in these notifications in the same way that you would use data returned by the order retrieval calls and [GetFeedback](#).

Note: Platform notifications are not identical in intent or content to the email messages that buyers and sellers might get when an item is listed, bid on, or purchased. However, they may contain some of the same data that appears in those messages.

To qualify for this feature, an application needs to be certified for use in the Production environment (see “[Compatible Application Check](#)” on page 67) and your membership tier needs to support eBay Platform Notifications. When these conditions are met, you can subscribe to eBay Platform Notifications by using functions in the API. The current release of the API does not directly support this functionality.

Unlike the API calls, which handle data using a request-response model, eBay Platform Notifications uses the notification style of operation. This is a one-way push from eBay. That is, you provide eBay with a URL, and then eBay delivers the notifications to you as activities occur. This means that if you subscribe to the Platform Notifications feature, you can reduce the number of times your application needs to execute order retrieval and feedback calls per day.

For information about working with eBay Platform Notifications, see “[Working with Platform Notifications](#)” on page 623.

Retrieving the Order Line Items for a Single Item

To retrieve the order line item data for one specified fixed-price item, use [GetItemTransactions](#). In general, only use this call when you cannot use [GetSellerTransactions](#).

Retrieving the order line item data for one item involves three general steps: setting up the execution environment, specifying the order line items to return (based on the listing’s item ID), and making the API call.

Retrieving the order line item data for a single item entails specifying the item ID for the item whose order line items are to be returned. Specify the item ID in the [ItemID](#) property of the call’s request .

Optionally, the call also takes a [TransactionID](#) or an [OrderLineItemID](#) as input to retrieve the data for a specific order line item associated with the specified listing. However, this practice results in a larger number of calls for the listing as a whole.

As this call can potentially return a very large number of order line items, use a time filter to limit the volume of the response to a particular window (range) of time. For more information, please see below and see the following topic: “[GetItemTransactions](#)” on page 836.

If you use the [NumberOfDays](#) time filter, the eBay database is queried to determine when order line items were last modified. Note that the modification status is updated each time the [TransactionArray.Transaction.Status.CheckoutStatus](#) value of an order line item changes. The checkout status can change any time (i.e., in seconds, minutes, or days) after the order line item is created. See “[Checkout](#)” on page 407.

Some order line items are processed with back-end batch jobs. This means there can be a short delay between an end time of an item and the recording of the initial order line item entry in the eBay database. Therefore, to make sure you do not miss any order line items, you should add time buffers, even if they are only a few seconds in length. Better yet, use the current date as the most recent date in the range to make sure you have captured all possible order line items and any updates. For each item, use [GetItemTransactions](#) at set intervals to get updates on any order line items that are new or have changed status since the last time you executed [GetItemTransactions](#) for that item.

Depending on the window of time you specify, the call can still return hundreds of order line items. Therefore, it is recommended that you configure the request to get one page of data at a time (where each page contains a more manageable number of order line items). Pagination breaks up the total data set into smaller chunks.

For example, if 100 order line items were generated for an item within the time that you specified, you could make one call for all 100 order line items, or ten calls with 10 order line items each (ten pages). Pagination is controlled by specifying the number of order line items to return in each page of data (i.e., for each call) and the page number to return.

The first call would return page 1 of the item's order line items. The [HasMoreTransactions](#) flag in the call response will indicate whether there are additional pages of order line items to retrieve. If there are additional pages to retrieve, you will see [HasMoreTransactions=1](#). The application can use that information to determine whether additional calls are needed and, based on the page size (the number of order line items per page), how many additional calls need to be made for all of the order line items to be retrieved. If additional calls are needed, each subsequent call would specify an incrementally higher page number to retrieve each succeeding page of order line items. If the [Pagination](#) container with the [EntriesPerPage](#) and [PageNumber](#) fields are not specified on input, no pagination occurs and all order line items (up to 100 maximum) are returned on a single page.

Retrieving the Order Line Items for a Specific Seller

To retrieve the order line item data for a specified seller, you can use [GetOrders](#) (see “[Best Practice for GetOrders](#)” on page 417), or alternatively, [GetSellerTransactions](#). Using [GetSellerTransactions](#) involves three general steps: setting up the execution environment, specifying the order line items to return, and making the API call.

An eBay user cannot use [GetSellerTransactions](#) to retrieve another seller's list of order line item data. eBay uses the security credentials (see “[Security](#)” on page 101) to determine whose order line item data to return.

This call can potentially return a very large number of order line items, so for each seller, use a time filter to limit the volume of the response to a particular window (range) of time. You need to execute the call at set intervals to retrieve updates on any order line items that are new or have changed status since the last time you executed [GetSellerTransactions](#). Please see below and see the following topic: “[GetSellerTransactions](#)” on page 865.

If you use the [NumberOfDays](#) time filter, the eBay database is queried to determine when order line items were last modified. Note that the modification status is updated each time the Checkout status of an item changes. The Checkout status can change any time (i.e., in seconds, minutes, or days) after the order line item is created. See “[Checkout](#)” on page 407.

Some order line items are processed with back-end batch jobs. This means there can be a short delay between an end time of an item and the recording of the initial order line item entry in the eBay database. Therefore, to make sure you do not miss any order line items, you should add time buffers, even if they are only a few seconds in length. Better yet, use the current date as the most recent date in the range to make sure you have captured all possible order line items and any updates. For each item, use the call at set intervals to get updates on any order line items that are new or have changed status since the last time you executed the call.

Depending on the window of time you specify, the call can still return hundreds of order line items. Therefore, it is recommended that you configure the request to get one page of data at a time (where each page contains a more manageable number of order line items). Pagination breaks up the total data set into smaller chunks. For more information, see the pagination information in the following topic: ["Retrieving the Order Line Items for a Single Item"](#) on page 418.

Understanding Buyer Payment

An integral part of the sale of an item is the buyer paying the seller for the order. How this is done depends in part on the payment methods the seller offers to the buyer (specified at listing time) and which one of these payment methods the buyer actually chooses.

A buyer may decide to pay for an order using PayPal (assuming it was one of the seller's payment methods). If this occurs, PayPal will handle the payment and set the Checkout status to complete.

Payment through PayPal works this way: If a buyer decides to pay using PayPal, the buyer will be redirected to the PayPal site at the end of the checkout process. The buyer pays on the PayPal site. Then, PayPal transfers the funds for the buyer's payment to the seller's PayPal account. The seller's PayPal account is identified by the PayPal email address that the seller specified at listing time. For general information about PayPal, see ["PayPal"](#) on page 153.

Note: The new payment process for the German (DE) and Austrian (AT) sites has been put on hold indefinitely. In this new payment process, new sellers will only be able to specify PayPal as the payment method, and buyers actually pay eBay and not the sellers for an order. After a holding period, eBay releases the funds to the sellers' accounts.

If a buyer selects a payment method other than PayPal, then arranging payment for the item is the seller's responsibility. If the payment method is an online payment service other than PayPal, then completing the payment may be a matter involving the seller, the buyer, and the payment service vendor. In these cases (when PayPal is not used), the status of an order is marked as completed as soon as the buyer finishes the checkout process. Note that this is true whether or not payment has actually been received by the seller, since that is outside the ability of eBay to track. Use [ReviseCheckoutStatus](#) to modify the status of an order. See ["ReviseCheckoutStatus"](#) on page 916.

After the item has been paid for and shipped, the seller can also use [CompleteSale](#) (see ["CompleteSale"](#) on page 792) to update the paid and shipped status in My eBay. For Selling Manager Pro users, the paid and shipped status and time stamps are returned in order line item-retrieval calls like [GetSellerTransactions](#).

Note: Once the new eBay payment process for sellers on the German (DE) and Austrian (AT) sites is rolled out, these sellers will not be able to mark an item as paid in My eBay or through the Paid field in the CompleteSale call.

Holds on PayPal Payments

Under certain conditions, a buyer's PayPal payment (see "Understanding Buyer Payment" on page 420) may be held by PayPal. Additional information is available on the PayPal site and at the following location:

When will my PayPal payment be available?
<http://pages.ebay.com/help/pay/questions/pending.html>

This section describes how to determine if a buyer's PayPal payment is held and, if it is, the type of hold that has occurred.

["Determining the Hold Status" on page 421](#)
["Payment Review Hold" on page 422](#)
["Merchant Hold" on page 423](#)
["New Seller Hold" on page 423](#)

Determining the Hold Status

The hold status of an item, and thus any payment hold that applies, is returned in the **Status.PaymentHoldStatus** field of the **Transaction** containers.

The possible values include:

- **MerchantHold**—Results from a possible issue with a seller.
- **NewSellerHold**—Results from a "new seller hold," in which PayPal may hold the payments to a new seller for up to 21 days. Sellers are considered new until they have met the following three criteria below:
 - More than 90 days have passed since first successful sale
 - More than 25 domestic sales
 - More than \$250.00 in total sales
- **None**—No payment hold applies to the item.
- **PaymentHold**—A payment hold has been placed on the item.
- **PaymentReview**—Results from a possible issue with a buyer.
- **ReleaseConfirmed**—The funds are available in the seller's account.
- **Released**—A payment hold on the item was released.
- **ReleasePending**—The process for the release of funds to the seller's account has been initiated.

The responses of the following calls include the **Status.PaymentHoldStatus** field:

- GetItemTransactions
- GetSellerTransactions
- GetOrderTransactions
- GetOrders
- GetMyeBayBuying
- GetMyeBaySelling
- GetSellingManagerSaleRecord
- GetSellingManagerSoldListings

Payment Review Hold

This hold results from a possible issue with a buyer. For more information, see the following:

Payment Review

<https://www.paypal.com/us/cgi-bin/webscr?cmd=xpt/cps/securitycenter/general/PaymentReviewOverviewFAQ-outside>

A description of payment review also is available in the Receiving Money section of the PayPal User Agreement, available through a keyword search on the PayPal site for “PayPal User Agreement.”

PayPal

<http://www.paypal.com>

You can use several Trading API calls (see “[Holds on PayPal Payments](#)” on page 421) to determine if a payment hold results from a review of a buyer payment.

In the Trading API, if a hold occurs based on payment review, then in the **Status** container, a **PaymentHoldStatus** of **PaymentReview** is returned, as in the following example:

Example 22-1 Buyer Payment Hold in Status Container

```
...  
<Status>  
...  
  <PaymentHoldStatus>PaymentReview</PaymentHoldStatus>  
...  
</Status>  
...
```

If the value in the **PaymentHoldStatus** field is **Released**, the hold was removed. If the value in the **PaymentHoldStatus** field is **None**, then there is no hold.

Merchant Hold

The “merchant hold” results from a possible issue with a seller, but is different from the type of hold described in “[New Seller Hold](#)” on page 423.

In a small percentage of cases where it has been determined the risk of dissatisfied buyers is higher, PayPal may delay release of payment to the seller until the buyer has left a positive feedback, or 21 days have passed without a dispute, claim, chargeback, or reversal filed on the transaction.

You can use several Trading API calls to determine if a merchant hold has occurred.

In the Trading API, if a hold occurs based on the merchant, then within the **Status** container, a **PaymentHoldStatus** of **MerchantHold** is returned, as in the following example:

Example 22-2 Merchant Hold in Status Container

```
...  
<Status>  
...  
  <PaymentHoldStatus>MerchantHold</PaymentHoldStatus>  
...  
</Status>  
...
```

If the value in the **PaymentHoldStatus** field is **Released**, then a hold was removed. If the value in the **PaymentHoldStatus** field is **None**, then there is no hold.

New Seller Hold

Under the “new seller hold,” PayPal may hold the payments to new sellers for up to 21 days. Money may be released faster if tracking information or delivery confirmation is provided, or if the buyer leaves positive feedback. Thus, the new seller hold is different from the type of hold described in “[Merchant Hold](#)” on page 423.

eBay defines a new seller account as one with less than 90 days since the first successful sale of an item.

In the Trading API, if a “new seller hold” occurs, then in the **Status** container, a **PaymentHoldStatus** of **NewSellerHold** is returned, as in the following example:

Example 22-3 New Seller Hold in Status Container

```
...  
<Status>  
...  
  <PaymentHoldStatus>NewSellerHold</PaymentHoldStatus>  
...  
</Status>  
...
```

Handling Non-Paying Bidders

In some cases, the buyer might not complete the item's purchase (at all or to the satisfaction of all of the seller's requirements). When this occurs (and after a specified period of time has elapsed), a unpaid item situation exists and the seller can enter into a process to recover from the situation. For more on the Unpaid Item process see "[Unpaid Item Disputes](#)" on page 455.

CompleteSale API Call

The [CompleteSale](#) call can be used by the seller to do the following tasks:

- Change the paid status of an item using the [Paid](#) boolean flag. This action will not be available to DE and AT sellers subject to the new payment process.
- Change the shipped status of an item using the [Shipped](#) boolean flag
- Set the item's shipment tracking information using the [Shipment](#) container
- Leave feedback for the buyer using the [FeedbackInfo](#) container.

For more information, see "[CompleteSale](#)" on page 792.

Shipping the Item

After an item has been sold, the seller must send the item to the buyer. The costs for shipping the item (if not paid for by the seller) are included in the total price charged to the buyer for the item. For more information on calculating shipping costs, see "[Specifying Calculated Shipping](#)" on page 341.

During the checkout process, the buyer selects the shipping service from those specified by the seller. The shipping method the buyer selects is part of the data retrieved for the order line item, contained in its shipping details. The seller must then use that shipping method to send the item to the buyer.

If during the checkout process, the buyer selects PayPal as the payment method for the item, the buyer is redirected to the PayPal site to effect the payment. There, the buyer may purchase a shipment for the item and the shipping method for this shipment may be different from any the seller specified and may be different from the shipping method selected during the checkout process on the eBay site. (Buyers familiar with the process will typically do one or the other, but not both. However, less experienced buyers might select a shipping method on the checkout page, not knowing that there can also be a shipping process on the PayPal site. Thus applications have to take this possible difference into consideration.)

If the seller has a PayPal account and the item was paid for using PayPal, the seller may use a feature on the PayPal site to print the mailing labels for the item's shipping. When the seller purchases mailing labels, a shipment record is created for the item. This shipment record can also be used in tracking the shipment (by the seller or the buyer) on the PayPal site.

After the item has been paid for and shipped, the seller can use [CompleteSale](#) (see “[CompleteSale](#)” on page 792) to update the paid and shipped status in My eBay. If shipment tracking is used (see the preceding paragraph), you can modify or delete the shipment tracking details in CompleteSale as well. For details, see the Call Reference entry for [CompleteSale](#). For Selling Manager Pro users, the paid and shipped status and time stamps are returned in order line item-retrieval calls like [GetSellerTransactions](#).

Final Value Fees and Credits

When an item is sold, or ends with a winning bid, or is purchased, you are charged a Final Value Fee (FVF). This fee is in addition to the listing fee that was charged when you listed the item. The total cost of selling an item is the listing fee plus the Final Value Fee.

The Final Value Fee applies whether or not you complete the sale with the buyer. The Final Value Fee is returned in all order retrieval calls.

Note: Traditionally, the calculation of the Final Value Fee has been based on the transaction price only. As of April 2011, the calculation of the Final Value Fee for non-store, auction listings in North American sites is based on total item cost, including shipping, handling and shipping insurance. Any sales tax charged to the buyer is excluded from the calculation. See the the Final Value Fee help page (<http://pages.ebay.com/help/sell/fvf.html>) for more information on how Final Value Fees are calculated.

The output of [AddItem](#) includes fees the seller incurs for the listing. Those fees don't include the Final Value Fee, which cannot be calculated until the item is sold. Those fees also don't account for credits (such as reserve fee refunds or eBay promotional credits).

To retrieve Final Value Fees, pass [IncludeFinalValueFee](#) with a value of [True](#) in [GetItemTransactions](#), [GetSellerTransactions](#), [GetOrders](#), or [GetOrderTransactions](#). The Final Value Fee is returned in [Transaction.FinalValueFee](#).

Note: For non-store, North American auction listings, the [FinalValueFee](#) value may change upon checkout (payment).. The initial Final Value Fee will be based on the Transaction price. The final Final Value Fee will be updated based on the additional shipping/handling/insurance costs.

Under certain circumstances, sellers may request a Final Value Fee Credit. For example, a seller can request full credit if a buyer backs out and does not buy an item. Fee credit values are returned by [GetAccount](#). Unlike Final Value Fees, however, credits cannot be reported at the order line item level. See “[Unpaid Item Disputes](#)” on page 455 for more information about fee credits.

Due to data processing and storage limitations and performance considerations, neither Final Value Fees nor fee credits are aggregated and reported for items as a whole.

See the Fees Overview on the eBay Web site for information about fees and fee credits.

Fees Overview
<http://pages.ebay.com/help/sell/fees.html>

For site-specific fee information refer to “[Fees per Site](#)” on page 594.

Leaving and Getting Feedback

At the completion of checkout, each user is expected to leave feedback for their order partner. The seller may leave feedback about the buyer and vice versa. While leaving feedback is completely optional, it is in every user's best interest to leave feedback after a successful item sale. The higher the feedback score a seller has, the greater the desirability of a prospective buyer to do business with that seller. A user who gets a reputation for regularly concluding sales without leaving feedback may find that other users with whom they do business may not feel obligated to leave feedback. While this does not negatively affect a user's feedback score, neither does it raise the score.

For basic information on feedback, see [VerifyEligibilityOnly](#). For information on implementing leaving and retrieving feedback in an application by using [LeaveFeedback](#) and [GetFeedback](#), see "[Leaving and Retrieving Feedback](#)" on page 432. As an alternative to [LeaveFeedback](#), you can use [CompleteSale](#) to leave feedback at the same time that you mark the item as paid and shipped (see "[CompleteSale](#)" on page 792).

Making Second Chance Offers for Items

Second Chance Offer is a feature that enables a seller to sell another item of the same type to one or more of the non-winning bidders in an auction. Second Chance Offer also enables a seller to offer an item to another non-winning bidder when the winning bidder in the original auction failed to complete the purchase of the item.

If you list items on the eBay Motors site, also see "[AddTransactionConfirmationItem](#)" on page 791.

When an auction ends successfully with a winning bidder, the auction may also have had one or more users who bid on the item, but were not the winning bidder. In this group of bidders, the seller has a known group of users who were interested in the item. If the seller has another of the same item available, the seller can offer this second item directly to one of the non-winning bidders in the auction. If the seller has multiple items, the seller may make second chance offers to multiple non-winning bidders. This allows the seller to quickly sell the second item without having to create a new listing (and hoping that the non-winning bidders see it and bid on it) and the item going through a complete competitive bidding process. By making the offer directly to a user who has already expressed obvious interest in the item, the seller has a better chance of selling the second item—and in much less time than a full new auction would take.

Although the ideal outcome of a successful auction is that the user who committed to purchase the item actually does so, instances do occur where the winning bidder either fails to follow through on the purchase commitment or outright refuses to do so. Rather than the seller being stuck with the unpurchased item and having to relist it (and having to go through the time it takes for a new auction to run its course), the Second Chance Offer feature allows the seller to offer the item directly to one or more of the bidders in the original auction who were not the winning bidder.

Extending a second chance offer for an item is entirely voluntary. Even though an item is eligible for a second chance offer, the seller may opt to relist the item in a regular auction that is open to all eBay users. Or the seller may choose to take no further action on the item (that is, make no further effort to sell the item). The seller may extend a second chance offer to one of the non-winning bidders from the original listing or all of the non-winning bidders. Each non-winning bidder from the original

listing may be the recipient of one second chance offer. If the seller has only that one item, then the seller may make one second chance offer to one of the non-winning bidders from the original listing. If the recipient of this offer does not accept it and purchase the item, the seller may then extend a second chance offer for the item to a different non-winning bidder. But if the seller has more of the same item or of an equivalent item, the seller may make second chance offers to as many of the non-winning bidders as the number of items the seller has. The seller might extend second chance offers to all non-winning bidders at once. Or the seller might extend an offer only to one non-winning bidder, waiting to see whether that recipient user buys the item before extending an offer to another non-winning bidder.

Second Chance Offer listings are only available for purchase to the qualified user to whom the seller extended the offer. A Second Chance Offer listing does not appear in any search or listings pages. The recipient of a second chance offer is notified of the offer via email, with a link directly to the Second Chance Offer listing. A Second Chance Offer listing has a limited duration (specified by the seller). If an offer recipient decides to purchase the item, the Second Chance Offer listing ends and the item is sold. If an offer recipient decides not to purchase the item and the listing's duration expires, the seller may opt to extend another Second Chance Offer to another one of the original listing's non-winning bidders (as long as that other non-winning bidder has not already been given a Second Chance Offer). A Second Chance Offer listing ends when the seller cancels it (before the offer recipient commits to purchasing it), when the specified offer duration expires, or when the offer recipient decides to purchase the offered item.

Supporting API Calls

The Second Chance Offer feature is implemented in the API through the dedicated calls listed below. It also includes two fields for items: [SecondChanceEligible](#) (part of the [SellingStatusType](#) class) and [OriginalItemId](#) (part of [ListingDetailsType](#)), returned in the result sets of other, general purpose calls that return item data (such as [GetItem](#), [GetSellerList](#) and [GetBidderList](#)).

Dedicated functions:

- [AddSecondChancelItem](#)
- [VerifyAddSecondChancelItem](#)
- [GetAllBidders](#)

Determining Item Eligibility for a Second Chance Offer

For the seller to use the Second Chance Offer feature for an item, that item must meet certain eligibility requirements. These requirements are:

Table 22-1 Second Chance Offer Item Eligibility Criteria

Criteria

Table 22-1 Second Chance Offer Item Eligibility Criteria (Continued)

Criteria
If the item is neither an eBay Motors item nor in some categories on U.S. and international sites for high-priced items (such as items in many U.S. and Canada Business and Industrial categories):
<ul style="list-style-type: none"> ■ The original auction ended successfully. ■ The item was sold to a winning bidder. ■ There was at least one non-winning bidder in the auction. ■ The seller has other instances to sell of the same item offered in the original listing.
If the item is an eBay Motors item or in some categories on U.S. and international sites for high-priced items (such as items in many U.S. and Canada Business and Industrial categories):
<ul style="list-style-type: none"> ■ The item must be a vehicle item (i.e., listed in one of the Passenger Vehicles, Motorcycles, or Other Vehicles categories) or in some categories on U.S. and international sites for high-priced items (such as items in many U.S. and Canada Business and Industrial categories). ■ The item was not listed in a Parts & Accessories category. ■ If the item ended without a winning bidder, the item is eligible for a second chance offer if: <ul style="list-style-type: none"> □ The item ended with the reserve price unmet (for items with a reserve price). □ There was at least one bidder. ■ If the item ended with a winning bidder, the item is eligible for a second chance offer if: <ul style="list-style-type: none"> □ There was at least one non-winning bidder. □ The seller has other, like items to offer.
For items (eBay Motors items and all other item types) where the winning bidder did not purchase the item (non-paying bidder):
<ul style="list-style-type: none"> ■ The auction ended with a winning bidder. ■ The winning bidder did not purchase the item. ■ There was at least one non-winning bidder in the auction.

A second chance offer can be made of items listed with a Buy-It-Now option if there was at least one non-winning bidder. That is, if the first bidder submitted a bid and the auction continued and ended as a normal, competitive bidding auction (i.e., standard Chinese auction), then the listing can be eligible for a second chance offer if it meets all other applicable criteria.

In an application, the eligibility of an item for a second chance offer is indicated by the `SellingStatusType.SecondChanceEligible` field, returned by calls that retrieve item data (such as `GetSellerList` and `GetItem`). An application uses one of these calls to determine that a particular item is eligible for a second chance offer before proceeding to the next step of actually making the offer.

The seller may extend as many second chance offers as there are non-winning bidders in the original listing. But the seller may extend a second chance offer only once per non-winning bidder. If the item is later re-listed in a new auction and again becomes eligible for a second chance offer, a bidder from the original listing may again receive a second chance offer if that user also bid in this second (re-list) auction.

Here are some other rules and considerations surrounding the Second Chance Offer feature:

- A seller may send multiple second chance offers for an ended listing—as long as the seller has duplicate items available. The number of second chance offers a seller makes cannot be more

than the number of items the seller has. If the seller has only one unique item, the seller can make only one second chance offer.

- A seller may make a second chance offer to a non-winning bidder up to 60 days after the original listing has closed.
- No Insertion Fees are charged for a second chance offer. Sellers will be charged a Final Value Fee if the second chance offer is accepted.

Extending a Second Chance Offer

Get a List of the Item's Bidders

The first step in extending a second chance offer is retrieving a list of all of the users who bid on the original listing. This is accomplished by a call to [GetAllBidders](#). One of the call modes for [GetAllBidders](#) returns a list of all of the original listing's bidders along with the ID of the winning bidder (if there was one) and the amount of the winning bid. From this list, the seller can select a single user to whom the seller will extend the second chance offer.

Make the Second Chance Offer

Once a list of the original listing's bidders has been retrieved, the seller can extend a second chance offer to one or more of those non-winning bidders. To do this, an application must use the [AddSecondChancelItem](#). Execute [AddSecondChancelItem](#) once for each target recipient non-winning bidder to which the seller wishes to extend the second chance offer.

Identify the recipient user by specifying the eBay user ID in the [RecipientBidderUserID](#) property of the [AddSecondChancelItem](#) request. A call to [AddSecondChancelItem](#) fails with an error if the user specified in [RecipientBidderUserID](#) has already received a second chance offer for the same item.

Specify the item ID for the item's original listing in the [OriginalItemId](#) property.

The application also passes a value in the [Duration](#) property that indicates how long (in days) the second chance offer is to be available to the specified recipient user. If the user does not elect to purchase the item in that time, the second chance offer expires. (The seller may also end the second chance offer before this duration has expired. For more on this see “[Terminating a Second Chance Offer](#)” on page 431.)

The price in the second chance offer can vary. If the original listing was an eBay Motors item and the auction ended without the reserve price met, the seller specifies the price in the [BuyItNowPrice](#) property of [AddSecondChancelItem](#). The price that the seller specifies must be less than or equal to the reserve price used for the original listing. The same is true in some categories on U.S. and international sites for high-priced items (such as items in many U.S. and Canada Business and Industrial categories). For all other items, the price is determined automatically by eBay and the application does not supply this value as input. It is the value of the highest bid that the specified offer recipient made on the item in the original listing. If that offer recipient declines to purchase the item and the seller makes another second chance offer to a different non-winning bidder, then the price is based on that bidder's highest bid.

A successful call creates a new second chance offer listing for the item (which includes assigning a new item ID to the item) and sends an email to the non-winning bidder from the original listing to whom the seller wishes to extend the second chance offer. The recipient user may then go to the

Second Chance Offer listing on the eBay site to purchase the item. The new Second Chance Offer listing is only available to the specified user. It does not appear in any search or listing page on the eBay site where someone who did not participate in the original listing could view or bid on the item.

[AddSecondChancelItem](#) returns data similar to that returned by [AddItem](#), including fees for the new listing. In addition, [AddSecondChancelItem](#) returns values that are specific to Second Chance Offer items:

- The item ID for the new offer listing ([ItemID](#))
- The user ID for the bidder to whom the second chance offer was extended ([RecipientBidderUserID](#))
- The start and end times of the second chance offer ([StartTime](#) and [EndTime](#), respectively)
- The fees for the new offer listing ([Fees](#))

Prior to making a using [AddSecondChancelItem](#) (which results in a new second chance offer listing), an application can test the input values and give the seller an idea of what the listing fees would be by using [VerifyAddSecondChancelItem](#). The [VerifyAddSecondChancelItem](#) call acts in every way like [AddSecondChancelItem](#), except that it does not actually create a new second chance offer listing or incur any actual listing fees. [VerifyAddSecondChancelItem](#) is to [AddSecondChancelItem](#) what [VerifyAddItem](#) is to [AddItem](#).

Second chance offers may only be made to users who have enabled the preference indicating they will accept such offers (see the "Preferences" tab on the My eBay page). If the input for [AddSecondChancelItem](#) specifies a user who has this preference set to not receive offers, the call fails with error 974, "Bidder not eligible to receive a Second Chance Offer." (Verbose error string: "Unfortunately, the bidder has chosen not to receive Second Chance Offer emails.")

The example below shows [AddSecondChancelItem](#) being used to extend a second chance offer to the user named "JimBob347." The length of time the offer will be open to that user (i.e., the duration of the second chance offer listing) is ten days. The original listing, the item from which went unsold, is "1974579415." Prior to making this call, the seller had retrieved a list of the non-winning bidders from the listing and selected the user "JimBob347" after verifying that this user has enabled the option to receive second chance offers.

Retrieving Item Data for Second Chance Offer Listings

Once a second chance offer listing has been created, it can be viewed by the seller or the offer recipient on the eBay site or through an API call that retrieves item data (such as [GetSellerList](#) or [GetItem](#)). If the listing's data is retrieved via one of these calls, all of the usual fields are returned, plus the item ID for the original listing (in the [ListingDetails.OriginalItemID](#) property of the [Item](#) object in the response object). In result sets for such functions as [GetSellerList](#) and [GetSellerEvents](#) where data for multiple items is returned, a second chance offer listing can be differentiated from regular auctions by the listing type as indicated in the [Item.ListingType](#) property. For second chance offer listings, the type is [PersonalOffer](#).

For more on retrieving data for item listings, see "[Retrieving Items](#)" on page 502.

Terminating a Second Chance Offer

The seller may cancel a second chance offer. This is done using [EndItem](#), passing the item ID for the second chance offer listing (not the item ID for the original listing). A second chance offer listing may be cancelled any time prior to the expiration of the listing's duration or the offer's recipient electing to purchase the item through the second chance offer listing.

For more information on using [EndItem](#) to end item listings, see "[Ending Items Early](#)" on page 389.

Chapter 23

Leaving and Retrieving Feedback

Feedback is a valuable indicator of a user's reputation on eBay.

You can use calls that are analogous to eBay site pages to enable an end-user to leave and retrieve feedback. [LeaveFeedback](#) ("LeaveFeedback" on page 903) enables a user to leave one feedback associated with a single order line item. The call must be made for each feedback the user leaves, so multiple feedbacks require multiple [LeaveFeedback](#) calls. [GetFeedback](#) ("GetFeedback" on page 831) retrieves a summary of feedback data or summary data plus detailed feedback entries.

A seller can retrieve their detailed seller ratings if the Detailed Seller Ratings feature is available on their site. For more information, see "[Detailed Seller Ratings](#)" on page 433.

After the successful end of an auction (or fixed-price sale), the participants are strongly encouraged to leave feedback about each other.

You cannot use the API for mutual feedback withdrawal, even though this functionality is available on the eBay site.

- ["Introduction to Feedback" on page 432](#)
- ["Detailed Seller Ratings" on page 433](#)
- ["Leaving Feedback for Another User" on page 435](#)
- ["Getting Feedback Left By Another User" on page 436](#)
- ["Responding to Feedback" on page 440](#)

Introduction to Feedback

Only sellers and winning bidders can leave or receive feedback. If multiple items were sold, the seller can leave feedback for each buyer and each buyer can leave feedback about the seller. Users with a reputation of not leaving feedback might find others reluctant to leave feedback for them.

Buyers and sellers can receive feedback score for repeat order line items if the order line items occur in two separate calendar weeks. Definition of a calendar week is Monday to Sunday, eBay San Jose time (i.e. Pacific Standard Time). Also, feedback scoring is based on order line item end time and not feedback left time. A member can impact another member's feedback score at most +1 or -1 in a week. Repeat positive or negative feedbacks in a week will not be counted, even if they have more than one order line items between them in that week.

When leaving feedback, a buyer rates an order experience as positive, negative, or neutral. A seller can only rate an order as positive or neutral (they can not give a buyer a negative rating). The user can add a comment explaining the score or expanding on it. For information about the corresponding API functionality, see "[Leaving Feedback for Another User](#)" on page 435.

Feedback ratings are aggregated into a summary score that is displayed next to the user ID, e.g. on the View Item page, and returned by the eBay API (see “[Getting Feedback Left By Another User](#)” on page 436). For the US site, extensive information about feedback is available in help topics, such as the following:

Feedback

<http://pages.ebay.com/help/feedback/index.html>

Feedback Forum

<http://pages.ebay.com/services/forum/feedback.html>

Feedback Policies: Overview

<http://pages.ebay.com/help/policies/feedback-ov.html>

Feedback Star Glossary

<http://pages.ebay.com/help/feedback/reputation-stars.html>

The higher a seller or buyer’s summary feedback, the more likely others will want to do business with the seller or buyer. However, users take into consideration the length of time a user has been an eBay member. For example, a user with a score of 10 who has only been a member for six months is different from a user with the same score who has been a member for six years.

A user can reply to feedback left by another party. You can provide this functionality to users with the **RespondToFeedback** call.

GetFeedback retrieves the feedback multiple users have left for a specified user. Neither the API nor the eBay site enable a user to retrieve all the feedbacks the user has left for other users.

Note: Due to the internal eBay mechanisms that process incoming feedback, there may be a delay of up to 30 minutes between the time that feedback is left and when it can be viewed. This affects leaving feedback both through the eBay site and the API, as well as viewing feedback through either the site or the API.

The eBay site enables a user to retract a feedback. This cannot be done using the API.

Detailed Seller Ratings

When buyers leave an overall Feedback rating (positive, neutral, or negative) for a seller, they also can leave ratings in four areas: item as described, communication, shipping time, and charges for shipping and handling. If a seller has these detailed ratings, they are displayed in the seller’s Feedback Profile and can be retrieved using the API (see “[Using the API for Detailed Seller Ratings](#)” on page 434). Buyers do not receive detailed ratings.

The Detailed Seller Ratings (DSR) feature is available all API-enabled country sites.

Sellers retrieve detailed ratings as averages of the ratings left by buyers. Additional information about detailed seller ratings is available on the eBay sites on which the Detailed Seller Ratings feature

has been implemented. For example, please see the following information about detailed seller ratings:

Coming Soon: Feedback 2.0
<http://pages.ebay.co.uk/services/forum/fb-main.html>

Detailed seller ratings provide granular ways for good sellers to differentiate themselves to buyers. The detailed seller rating system is based on a one- to five-star scale. Five stars is the highest rating. Detailed seller ratings do not affect the overall feedback score.

Using the API for Detailed Seller Ratings

In the Detailed Seller Ratings feature, the [LeaveFeedback](#) (“[LeaveFeedback](#)” on page 903) and [GetFeedback](#) (“[GetFeedback](#)” on page 831) calls can include fields that contain detailed seller ratings data. For site availability of the feature, see “[Detailed Seller Ratings](#)” on page 433.

Specifically, [LeaveFeedback](#) input can include a [SellerItemRatingDetailArray](#) container with detailed seller ratings, as in the following example:

Example 23-1 Sample XML for Leaving Seller Ratings with LeaveFeedback

```
...
<SellerItemRatingDetailArray>
  <ItemRatingDetails>
    <RatingDetail>ItemAsDescribed</RatingDetail>
    <Rating>3</Rating>
  </ItemRatingDetails>
  <ItemRatingDetails>
    <RatingDetail>Communication</RatingDetail>
    <Rating>4</Rating>
  </ItemRatingDetails>
  <ItemRatingDetails>
    <RatingDetail>ShippingTime</RatingDetail>
    <Rating>2</Rating>
  </ItemRatingDetails>
  <ItemRatingDetails>
    <RatingDetail>ShippingAndHandlingCharges</RatingDetail>
    <Rating>1</Rating>
  </ItemRatingDetails>
</SellerItemRatingDetailArray>
...
```

A [GetFeedback](#) call can return [FeedbackSummary.SellerAverageRatingDetailArray](#), which contains average detailed seller ratings. The following shows an example of a [SellerAverageRatingDetailArray](#) container:

Example 23-2 Sample SellerAverageRatingDetailArray Container of GetFeedback

```
...
<SellerAverageRatingDetailArray>
  <AverageRatingDetails>
    <RatingDetail>ItemAsDescribed</RatingDetail>
    <Rating>3.26</Rating>
    <RatingCount>40</RatingCount>
  </AverageRatingDetails>
  <AverageRatingDetails>
    <RatingDetail>Communication</RatingDetail>
    <Rating>4.52</Rating>
    <RatingCount>53</RatingCount>
  </AverageRatingDetails>
  <AverageRatingDetails>
    <RatingDetail>ShippingTime</RatingDetail>
    <Rating>2.0</Rating>
    <RatingCount>53</RatingCount>
  </AverageRatingDetails>
  <AverageRatingDetails>
    <RatingDetail>ShippingAndHandlingCharges</RatingDetail>
    <Rating>1.0</Rating>
    <RatingCount>53</RatingCount>
  </AverageRatingDetails>
</SellerAverageRatingDetailArray>
...
```

Additionally, as part of the Detailed Seller Ratings feature, a [GetFeedback](#) call can return [FeedbackDetailArray.FeedbackDetail.ItemID](#) (the ID for the relevant item listing) and [FeedbackDetailArray.FeedbackDetail.ItemPrice](#) (the final price of the relevant item listing). This information can provide buyers with more information for evaluating a seller's reputation. However, these fields are returned for an item only if necessary conditions are met, e.g. the feedback was left within the past 90 days, the item was not for mature audiences, and the item was not a private listing.

Leaving Feedback for Another User

You can enable a “requestor” (seller or buyer) to leave feedback by using [LeaveFeedback](#). There must be an order relationship between the users.

That is, the requestor and recipient must have been participants in a successfully concluded listing (concluded no more than 60 days before the attempt to leave feedback). Whichever role the requestor has (seller or buyer), the recipient must have been in the opposite role.

Identify the user about whom the feedback is being left in the [TargetUser](#) property of [LeaveFeedback](#). This property takes a value of type [UserIDType](#) that is the eBay user name (also called the eBay user ID) for the recipient.

To indicate the order relationship between the requestor and the recipient, specify the item ID for the successfully concluded listing in the **ItemID** property.

If the listing offered only a single item (i.e., an auction listing or a single-item, fixed-price listing), this item ID is enough to identify the order line item associating the two users. This is because such listings can only spawn one order line item. But a multi-item, fixed-price listing has the potential for spawning multiple order line items, up to the number of items offered in the listing. In these cases, the application must specify the ItemID/TransactionID pair or the OrderLineItemID for that particular order line item. Specify the TransactionID in the **TransactionID** property, which is of type **string**. **TransactionID** is optional, unless the listing for the feedback is a multi-item listing when it is a required input.

The feedback itself consists of a score and a comment that expands on or justifies the score. Specify the feedback score—positive, neutral, or negative—in the **CommentType** property. **CommentType** is of type **CommentTypeCodeType**, with valid input values of **Positive**, **Neutral**, and **Negative**. Specify a textual comment for the feedback in the **CommentText** property (type **string**).

LeaveFeedback can be used to leave one feedback. It cannot be used to leave multiple feedbacks, even when the requestor is the seller in a multi-item listing and the recipients all buyers from that same listing. To leave multiple feedbacks, an application needs to use **LeaveFeedback** once for each feedback.

Some feedback restrictions apply to protect members from receiving false or inaccurate feedback from other members:

- Sellers are not able to leave negative or neutral feedback for buyers.
- Buyers are not able to leave neutral or negative feedback for sellers within 7 days of the order line item creation.
- Any neutral or negative feedback left by suspended members' will not count.
- Any neutral or negative feedback left by a buyer who has not responded to a UPI dispute will not count.

An application can give the appearance of letting the end-user leave multiple feedbacks in batches. This is done by allowing the end-user to enter the data for as many feedbacks as desired, perhaps store the information in a database. When the user clicks a button, **LeaveFeedback** is executed once for each pending feedback.

The result set of **LeaveFeedback** includes **FeedbackID**, which is the ID of the feedback that has been left. The response object also contains the fields “inherited” from the abstract response object, such as the field with the date and time the call was made.

Getting Feedback Left By Another User

Applications can use **GetFeedback** to enable users to retrieve feedback left for them by others. For information about production use of this call, see “[Compatible Application Check for Specific Calls](#)” on page 963.

Making the Call

You can specify a valid eBay user ID in the **UserID** property of the **GetFeedback** request. This property is optional for specifying the user for whom to retrieve feedback. If no value is specified, feedback data is returned for the requesting user.

GetFeedback can only be used to retrieve feedback other users have left about the user specified in the **UserID** property (or the requesting user if no user ID is specified). It cannot be used to retrieve the feedbacks the requesting user left about other users.

The child elements of the **Pagination** property (which is a **PaginationType** object) control the number of feedback detail entries returned by any given call. (Use pagination only when the call specifies a detail level that returns feedback details.) Use the **Pagination.EntriesPerPage** property to specify the maximum number of feedback detail entries to return in any single call. A single call represents one “page” of data. Use the **Pagination.PageNumber** to specify which page of data to return. For example, if **Pagination.EntriesPerPage** has been set to 50, then to retrieve feedback detail entries 51-100 a value of 2 needs to be specified for **Pagination.PageNumber** in the current call. The default value for the **Pagination.EntriesPerPage** property is 25. An application may set **Pagination.EntriesPerPage** only to one of the values 25, 50, 100, or 200. A single call can only return a maximum of 200 feedback detail entries. If there are more than 200 feedback detail entries for the specified user, then multiple calls are needed to return all of them.

When pagination is used, an application needs to calculate how many calls need to be made in order to retrieve all of the feedback detail entries. At least one call needs to be made before this calculation can be performed, as it requires one of the values from the result set. After this first call, divide the number returned in the **FeedbackDetailItemTotal** property of the response object by the number specified in the **Pagination.EntriesPerPage** input property, adding one if the numbers are not evenly divisible. The result of this calculation is the number of calls that would be needed to retrieve every last feedback detail entry (and the summary data that comes with the last page). Make repeated calls, each with an incrementally higher value in the **Pagination.PageNumber** input property, starting with 1 and the last with a value derived from the calculation.

Working with the GetFeedback Result Set

Depending on the detail level you specify, **GetFeedback** returns a summary of the user’s feedback or a summary with detailed feedback entries.

Adding up the detailed feedback entries results in the summary scores and the total aggregate feedback score. The feedback summary does not include comments from individual feedbacks. Therefore a user may prefer detail entries.

By default, **GetFeedback** returns only the summary feedback data for the specified user. An application needs to use a detail level of **ReturnAll** to have the feedback details returned. When feedback details are returned, the feedback summary is returned on the final page of data.

Summary Feedback Data

After a successful call, the **FeedbackSummary** property of the **GetFeedback** object (which is of type **FeedbackSummaryType**) contains the summary feedback data for the specified user.

Summary feedback data contains several categories (see “[Elements of FeedbackSummaryType](#)” on page 438). Each category is represented as a property of the **FeedbackSummaryType** object. There also

are properties that return simple counts of types of feedbacks: neutral comments from suspended users, unique neutral feedbacks, unique negative feedbacks, and unique positive feedbacks.

Table 23-1 Elements of FeedbackSummaryType

Property	Purpose
BidRetractionFeedbackPeriodArray	Summary data for feedbacks for bids the specified user retracted.
NegativeFeedbackPeriodArray	Summary data for negative feedbacks.
NeutralFeedbackPeriodArray	Summary data for neutral feedbacks.
PositiveFeedbackPeriodArray	Summary data for positive feedbacks.
SellerAverageRatingDetailArray	Summary data for detailed seller ratings (see " Detailed Seller Ratings " on page 433)
TotalFeedbackPeriodArray	Summary data for all types of feedbacks.
NeutralCommentCountFromSuspendedUsers	Count of neutral feedbacks from suspended users.
UniqueNegativeFeedbackCount	Count of unique negative feedbacks, including weekly repeats.
UniquePositiveFeedbackCount	Count of unique positive feedbacks, including weekly repeats.
UniqueNeutralFeedbackCount	Count of unique neutral feedbacks, including weekly repeats.

Several categories of feedback summary data are in `FeedbackPeriodArrayType` objects. Each `FeedbackPeriodArrayType` object contains 3-4 `FeedbackPeriodType` objects. A `FeedbackPeriodType` object contains the count of feedbacks of the category of the `FeedbackPeriodArrayType` object of which it is a child.

For example, a `NeutralFeedbackPeriodArray.FeedbackPeriod` object contains a count of *neutral* feedbacks for the period the object covers. Each `FeedbackPeriodType` object contains both a feedback count for the period (the `FeedbackPeriodType.Count` property) and the span of the period in days (`FeedbackPeriodType.PeriodInDays`).

Under the current feedback scheme used by eBay, there are time periods for which a count of feedbacks received by the specified user are provided. The time periods are relative to the date and time of the call used to retrieve the feedback data.

- The 7 days preceding the call (`BidRetractionFeedbackPeriodArray` only)
- The 30 days preceding the call
- The 180 days preceding the call
- The 365 days preceding the call (except in `BidRetractionFeedbackPeriodArray`)

Note that for bid retraction feedbacks, summary data is provided for the following periods: 7, 30, and 180 days.

The table below maps the `FeedbackPeriodType` objects, their parent `FeedbackPeriodArrayType` objects, and the time intervals for each.

Table 23-2 Feedback Types and Periods

FeedbackPeriodArrayType Object	FeedbackPeriodType.PeriodInDays Value
<code>FeedbackSummary.BidRetractionFeedbackPeriodArray</code>	7
<code>FeedbackSummary.BidRetractionFeedbackPeriodArray</code>	30
<code>FeedbackSummary.BidRetractionFeedbackPeriodArray</code>	180
<code>FeedbackSummary.NegativeFeedbackPeriodArray</code>	30
<code>FeedbackSummary.NegativeFeedbackPeriodArray</code>	180
<code>FeedbackSummary.NegativeFeedbackPeriodArray</code>	365
<code>FeedbackSummary.NeutralFeedbackPeriodArray</code>	30
<code>FeedbackSummary.NeutralFeedbackPeriodArray</code>	180
<code>FeedbackSummary.NeutralFeedbackPeriodArray</code>	365
<code>FeedbackSummary.PositiveFeedbackPeriodArray</code>	30
<code>FeedbackSummary.PositiveFeedbackPeriodArray</code>	180
<code>FeedbackSummary.PositiveFeedbackPeriodArray</code>	365
<code>FeedbackSummary.TotalFeedbackPeriodArray</code>	30
<code>FeedbackSummary.TotalFeedbackPeriodArray</code>	180
<code>FeedbackSummary.TotalFeedbackPeriodArray</code>	365

The `Count` property of each `FeedbackPeriodType` object contains the number of feedbacks received in the period circumscribed by the value in the `FeedbackPeriodType.PeriodInDays` property. This model is then repeated 3-4 times (once for each time period) for most feedback categories.

The counts of feedbacks received during each of the time periods are for informational purposes only. These counts themselves cannot be used to calculate the user's total feedback score. This is because the counts in each period for a given feedback category are incrementally more inclusive. For example, assume a user has received 1 positive feedback in the past 30 days and 1 positive feedback in the past 31-180 days. The counts for this would be 1 for the 30-day period and 2 for the 180-day period: the positive feedback in the first 30 days is counted in the total for the 180-day period.

To programmatically calculate the user's total feedback score: subtract the count of unique negative feedbacks (returned in `UniqueNegativeFeedbackCount`) from the count of unique positive feedbacks (from `UniquePositiveFeedbackCount`). Of course, eBay does this calculation and the user's total feedback score is returned in the `GetFeedback` result set, in the `FeedbackScore` property of the `GetFeedback` object.

Detailed Feedback Data

The `FeedbackDetail` property contains the still-active individual feedback entries other users have left for the user. Each feedback in this array is represented by a `FeedbackDetailType` object. The `FeedbackDetailItemTotal` property of the `GetFeedback` object indicates the number of individual feedback detail entries that are returned in `FeedbackDetail` for the current call. If feedback detail entries are returned, the application needs to traverse the `FeedbackDetailType` objects contained in

the **FeedbackDetail** property, inspect the properties of each **FeedbackDetailType** object, and then use the data (per the use case of the application). For example, an application might display the feedback detail entries in a scrollable user interface control (like a data grid) where the end-user can view them. Or, it might only use them to programmatically calculate values or aggregate the data in an organization different from the feedback summary returned by [GetFeedback](#).

The **FeedbackDetailType** object contains the data for one feedback detail entry. There can be zero, one, or multiple **FeedbackDetailType** objects returned in the **FeedbackDetail** property for any given call. The data returned in one **FeedbackDetailType** object includes data submitted using [LeaveFeedback](#) (see “[Leaving Feedback for Another User](#)” on page 435 for more information on leaving feedback). Additionally, the **FeedbackDetailType** object contains information about whether eBay replaced the text of a feedback comment, response, or follow-up.

If the application is aggregating the user’s feedback scores, it can use the same logic as eBay uses. For this purpose: a positive feedback (**Positive** in the **CommentType** property) counts as 1 (one), a neutral feedback (**Neutral** in **CommentType**) counts as 0 (zero), and a negative feedback (**Negative** in **CommentType**) counts as -1 (negative one).

Responding to Feedback

By using [RespondToFeedback](#), applications can allow users to reply to feedback that has been left for them or post follow-up comments to feedback that they have left for other users. Replies to feedback are typically left in response to neutral or negative feedback, where the user for whom feedback has been left wants to provide an explanation for the non-positive feedback comment.

To use the call, the application must have either a **FeedbackID** (retrieved by using [GetFeedback](#)), the **ItemID/TransactionID** pair, or the **OrderLineItemID** for the item for which feedback has been left. In addition, the application must have the **UserID** for the user who left the feedback that is being replied to or followed up on.

You can not use [RespondToFeedback](#) to automatically leave responses to negative feedback, as this is not in accordance with eBay policy.

For more information on using this call, see “[RespondToFeedback](#)” on page 912.

Chapter 24

Disputes Overview

In this section:

- ["Resolving Selling Issues" on page 441](#)
- ["Resolving Buying Issues" on page 442](#)
- ["eBay Buyer Protection Program" on page 443](#)
- ["Legacy eBay Dispute System" on page 444](#)
- ["PayPal Buyer Protection" on page 444](#)
- ["Disputes and the Resolution Case Management API" on page 446](#)

In most cases, the purchase of eBay items through auctions or fixed-priced listings work great for both buyers and sellers. However, there are times when issues may arise between the buyer and seller.

Note: A buyer is committed to purchase an item once he/she either buys a fixed-priced item, accepts a second chance offer, uses the Buy It Now option in an auction, is the winning bidder in an auction, or makes a Best Offer that is accepted by the seller.

Communication between a buyer and seller to resolve an issue is always encouraged by eBay, and is often the fastest and simplest way to resolve a problematic transaction. However, if the buyer and seller on either side of a transaction are unable to resolve an issue, eBay sites provide an online process that allow a buyer or seller to file, respond to, and manage disputes.

Resolving Selling Issues

The primary issue that a seller can have with a buyer is when the buyer has committed to purchasing an item, but has not paid for the item in a timely manner. Below are a few options available to the seller before it is necessary for the seller to open an Unpaid Item case against the buyer:

- **Contact the buyer through My eBay:** give the buyer an adequate amount of time to respond to the email, and consider the possibility that the buyer may be experiencing a computer issue, or perhaps addressing another issue or emergency.
- **Use eBay's Find contact information tool:** if an email is unsuccessful, a seller can use eBay's [Find contact information](#) tool to obtain the buyer's information. This tool requires the buyer's user ID and the Item ID. After successfully using this tool, the buyer's contact information is emailed to the seller.
- **Send one or more payment reminders:** sometimes a buyer may only need a reminder that payment is due on an item. Payment reminders may be sent between three and 30 days after a listing ends (or a purchase is made in a fixed-price listing).

If none of the three methods listed above are successful, the seller can open an Unpaid Item case in the Resolution Center. The Resolution Center is only available on the US, UK, and DE sites. A seller is

allowed to open an Unpaid Item case between four and 32 days after a listing ends (or a purchase is made in a fixed-price listing). The buyer is notified immediately when an Unpaid Item case is filed, and given four days to respond and/or pay for the item. If the buyer doesn't respond and/or pay within four days, the seller has the option of closing the case and receiving credit for the Final Value Fee.

Once a case has been closed, the seller has the options of extending a Second Chance Offer to another bidder or relisting the item. The seller may be eligible for an insertion fee credit. See [Getting credited for the insertion fee](#) help topic for more information and eligibility requirements.

Unpaid Item disputes are discussed in detail in “[Unpaid Item Disputes](#)” on page 455.

Resolving Buying Issues

The two main issues that a buyer can have with a seller are summarized below:

- **Item Not Received:** the buyer has paid for the item, but the item has not been received within the expected timeframe.
- **Significantly Not As Described:** the buyer has received the item, but the item is damaged or it doesn't match the listing description.

eBay recommends that the buyer allows the seller to make things right before opening a case. Below are a few options available to the buyer before it is necessary for the buyer to open an eBay Buyer Protection case against the seller:

- **Contact the seller through My eBay:** give the seller an adequate amount of time to respond to the email, and consider the possibility that the seller may be experiencing a computer issue, or perhaps addressing another issue or emergency. It is a good idea for the buyer to investigate if email spam filters are blocking messages from the seller. Also, the buyer should check to see if their contact information is current by going to the **Account** tab in My eBay.
- **Use the Ask a question link:** if an email is unsuccessful, a buyer can click on the **Ask a question** link near the bottom of the item listing. The buyer may be able to find the appropriate question to ask the seller, or the buyer can select the **No, I want to contact the seller** option under **Did we answer your questions?**. See the [Contacting your seller](#) help page for more information about contacting the seller.

If the methods above are unsuccessful, the buyer can open an eBay Buyer Protection Item Not Received or Significantly Not As Described case in the Resolution Center. The Resolution Center is only available on the US, UK, and DE sites. A buyer is allowed to open an eBay Buyer Protection case up to 45 days after the purchase of the item. The seller is notified immediately when an eBay Buyer Protection case is filed against them, and given seven days (or 10 days for international transactions) to respond to the case, and work toward resolving the issue. After the allotted time, the buyer can escalate the case to eBay customer support.

For an Item Not Received case, resolution may involve the following:

- The seller providing proof of delivery to eBay. For items valued over \$250, signature confirmation is required.
- The seller coming to an agreement with the buyer to send the original item or a replacement item.
- The seller refunding the buyer the full cost of the item including any applicable sales tax and original shipping costs.
- The seller reimbursing eBay if eBay has already refunded the buyer after opening a case. If the buyer eventually receives the item, but outside of the seller's stated handling time, the buyer must return the item to the seller before receiving a refund.

For a Significantly Not As Described case, resolution may involve the following:

- The seller providing proof to eBay that the item was described accurately and consistently throughout the listing and all associated communication between the buyer.
- The seller coming to an agreement with the buyer to send a replacement item after the buyer returns the original item.
- The seller refunding the buyer up to the full cost of the item plus any applicable sales tax and original shipping costs. For a full refund to occur, it is generally required that the buyer return the item to the seller.

The two buyer-created disputes are Item Not Received and Significantly Not As Described. These dispute types are discussed in detail in ["Buyer Disputes"](#) on page 476.

eBay Buyer Protection Program

The eBay Buyer Protection program and the Resolution Center are available on the US UK, and DE sites. For listings on these sites, eBay hosts the dispute resolution process when buyers claim to a seller that their item was not received or the item they received is not as it was described in the eBay item listing. See ["Resolving Buying Issues"](#) on page 442 for tips on resolving buying issues. Sellers can file Unpaid Item disputes in Resolution Center. See ["Resolving Selling Issues"](#) on page 441 for tips on resolving selling issues.

The dispute process for other international eBay sites vary. Many sites still use a legacy dispute system to handle disputes. See ["Legacy eBay Dispute System"](#) on page 444 for more information. If an item is purchased on an eBay site using PayPal as the payment method, there is a possibility that the item will be covered by the PayPal Buyer Protection policy. See ["PayPal Buyer Protection"](#) on page 444 for more information.

For more information on the eBay Buyer Protection program, see the [eBay Buyer Protection help](#) page. Buyers may want to read the [Buyers' FAQs about the eBay Buyer Protection Policy](#), and sellers may want to read the [Sellers' FAQs about the eBay Buyer Protection Policy](#).

Legacy eBay Dispute System

For eBay sites that have not implemented the eBay Buyer Protection program and Resolution Center, the eBay legacy dispute resolution system is used. The dispute process between buyer and seller varies by site. For clarity, 'dispute console' is used to represent a Web interface where buyers and sellers can perform dispute actions.

PayPal Buyer Protection

Buyers who purchase an item on eBay using PayPal as the payment method may still file a dispute using PayPal Buyer Protection. However, once a PayPal Buyer Protection dispute is filed, the buyer may not file an eBay Buyer Protection dispute for the same order line item (transaction). Log in to your PayPal account and click on the Resolution Center link to get more information about the PayPal Buyer Protection program.

Overview of Dispute Actions

[Table 24-1](#) provides an overview of the dispute actions available to the buyer and seller on either side of an eBay order line item (transaction).

Table 24-1 Buyer and Seller Dispute Actions

Action	User Role	Site to perform action	API call to perform action
Create an Unpaid Item dispute	Seller	Resolution Center	Trading API - AddDispute
Cancel an unpaid transaction	Seller	Resolution Center	Trading API - AddDispute
Create an Item Not Received case	Buyer	Resolution Center or dispute console	None
Create a Significantly Not As Described case	Buyer	Resolution Center or dispute console	None
Retrieve summary information on all cases (all case types)	Seller or Buyer	Resolution Center or dispute console	Resolutions API - getUserCases (seller only)
Retrieve detailed information on a single eBay Buyer Protection Item Not Received or Significantly Not As Described case	Seller or Buyer	Resolution Center or dispute console	Resolutions API - getEBCaseDetail (seller only)
Retrieve detailed information on all disputes and cancelled transactions (except eBay Buyer Protection cases)	Seller or Buyer	Resolution Center or dispute console	Trading API - GetUserDisputes (seller only)
Retrieve detailed information on a single Unpaid Item dispute or cancelled transaction	Seller	Resolution Center or dispute console	Trading API - GetDispute
Respond to or close an Unpaid Item dispute on cancelled transaction case	Seller	Resolution Center or dispute console	Trading API - AddDisputeResponse
"Reverse" a closed dispute. In a "reversed" dispute, the seller's Final Value Credit and buyer's Unpaid Item Strike are both reversed	Seller	None	Trading API - SellerReverseDispute

Table 24-1 Buyer and Seller Dispute Actions

Action	User Role	Site to perform action	API call to perform action
Retrieve possible options on an eBay Buyer Protection Item Not Received or Significantly Not As Described case	Seller	Resolution Center	Resolutions API - getActivityOptions
Offer the buyer a full refund if the Significantly Not As Described item is returned (eBay Buyer Protection cases)	Seller	Resolution Center	Resolutions API - offerRefundUponReturn
Issue a full refund to the buyer (eBay Buyer Protection cases)	Seller	Resolution Center	Resolutions API - issueFullRefund
Offer a partial refund to the buyer (eBay Buyer Protection cases)	Seller	Resolution Center	Resolutions API - offerPartialRefund
Issue a partial refund to the buyer (eBay Buyer Protection cases)	Seller	Resolution Center	Resolutions API - issuePartialRefund
Provide shipping information to other party (eBay Buyer Protection cases)	Seller or Buyer	Resolution Center	Resolutions API - provideShippingInfo (seller only)
Provide package tracking information to other party (eBay Buyer Protection cases)	Seller or Buyer	Resolution Center	Resolutions API - provideTrackingInfo (seller only)
Provide an alternative address and a Return Merchandise Authorization number to other party (eBay Buyer Protection cases)	Seller	Resolution Center	Resolutions API - provideReturnInfo
Post a message to the buyer regarding a refund. (eBay Buyer Protection cases)	Seller (Germany only)	Resolution Center	Resolutions API - provideRefundInfo
Upload proof of shipping or proof of refund documents. (eBay Buyer Protection cases)	Seller (Germany only)	Resolution Center	Resolutions API - uploadDocuments
Offer another solution to buyer to help resolve and close (eBay Buyer Protection cases)	Seller	Resolution Center	Resolutions API - offerOtherSolution
Escalate an eBay Buyer Protection case	Seller or Buyer	Resolution Center	Resolutions API - escalateToCustomerSupport (seller only)
Appeal eBay's decision on an eBay Buyer Protection case	Seller or Buyer	Resolution Center	Resolutions API - appealToCustomerSupport (seller only)

Disputes and the Resolution Case Management API

In this section:

- “Overview” on page 446
- “Using Case Retrieval Calls” on page 448
- “Using Filters With the `getUserCases` Call” on page 449
- “Retrieving a List of Next Possible Actions on a Case” on page 450
- “Requesting the Return of a SNAD Item” on page 451
- “Issuing a Full Refund to the Buyer” on page 451
- “Offering a Partial Refund to the Buyer” on page 451
- “Issuing a Partial Refund to the Buyer” on page 451
- “Providing Shipping Information to the Buyer” on page 452
- “Providing Tracking Information to the Buyer” on page 452
- “Providing Return Information to the Buyer” on page 452
- “Providing Refund Information to the Buyer” on page 452
- “Uploading Proof Documents” on page 452
- “Offering Other Solution” on page 453
- “Escalating a Case” on page 453
- “Appealing a Case” on page 453
- “eBay Buyer Protection Platform Notifications” on page 454

Overview

The Resolution Case Management API is intended to be consumed by sellers who are interested in retrieving summary information on all dispute types, and managing and tracking eBay Buyer Protection Item Not Received and Significantly Not As Described cases. eBay Buyer Protection claims can only be filed through the US, UK, and DE Resolution Centers available at: <http://resolutioncenter.ebay.com/>, <http://resolutioncentre.ebay.co.uk/>, and <http://resolutioncenter.ebay.de/>.

The Resolution Case Management API contains the following calls:

- **`getUserCases`**: this call is used to retrieve all legacy, PayPal, and eBay Buyer Protection cases. This call does not require any input parameters, but there are four types of filters that help customize the cases that are returned. This call can only return cases created in the last 18 months. This call only retrieves summary information on each case, including the case ID. To retrieve full details and the current status of a legacy or PayPal case, the case ID must be passed into the Trading API's `GetDispute` call. To retrieve full details and the current status of an eBay Buyer Protection case, the case ID must be passed into the Resolution Case Management API's `getEBPCaseDetail` call.
- **`getEBPCaseDetail`**: this call is used to retrieve details on a single eBay Buyer Protection Item Not Received or Significantly Not As Described case. The case ID and case type (EBP_INR or

EBP_SNAD) for the case that is retrieved in the [getUserCases](#) call is required as input parameters for the [getEBPCaseDetail](#) call.

- [getActivityOptions](#): this call returns a list of the next possible actions that a seller can make on a case, including escalating a case, appealing a case decision, issuing a full refund to the buyer, providing tracking information to the buyer, and offering another solution to the buyer.
- [offerRefundUponReturn](#): this call is used by sellers to inform buyers that a Significantly Not As Described item must be returned before a full refund can be issued.
- [issueFullRefund](#): this call allows the seller to issue a full refund to the buyer in order to settle an open case.
- [offerPartialRefund](#): this call allows the seller to issue a full refund to the buyer in order to resolve an open Significantly Not As Described case.
- [issuePartialRefund](#): this call allows a seller to issue a partial refund to a buyer to help resolve an open Significantly Not As Described case. The partial refund amount must match what was offered and accepted by the buyer, and the currency type must match what was used in the original transaction. This call can only be used if the payment method is PayPal.
- [provideShippingInfo](#): this call allows the seller to provide the shipping carrier and shipped date to a buyer if the seller is sending the original item, a replacement item, or missing/replacement parts to the buyer.
- [provideTrackingInfo](#): this call allows the seller to provide tracking information to a buyer if the seller is sending the original item, a replacement item, or missing/replacement parts to the buyer.
- [provideReturnInfo](#): this call is used by the seller to provide an alternative shipping address to the buyer who is returning an item. In this case, an "alternative" shipping address is one other than the seller's default shipping address or return address on file for the seller.
- [provideRefundInfo](#): this call is used by a seller (Germany only) to provide a customized message to the buyer regarding an item refund.
- [uploadDocuments](#): this call is used by a seller (Germany only) to upload one to five documents that act as proof that an item was shipped or proof that an order was fully or partially refunded.
- [offerOtherSolution](#): this call allows the seller to offer the buyer a customized solution in order to settle an open case. Possible use cases for using this call including issuing a partial refund to the buyer, providing shipping or refund information to the buyer, or requesting that the buyer return a Significantly Not As Described item before a refund is issued.
- [escalateToCustomerSupport](#): this call allows the seller to escalate an open case to eBay customer support.
- [appealToCustomerSupport](#): this call allows the seller to make an appeal to eBay customer support after an unfavorable case decision.
- [getVersion](#): this call returns the current version of the Resolution Case Management service.

Using Case Retrieval Calls

The following table contains information on case types that can be retrieved with the Resolution Case Management API and the dispute calls ([GetUserDisputes](#) and [GetDispute](#)) of the Trading API. The table also includes enumeration values and site availability. These enumeration values (along with case ID values) are passed into the [getEBPCaseDetail](#) (Resolution Case Management API) and [GetDispute](#) (Trading API) calls to retrieve detailed information on specific cases.

Table 24-2

API - Call	Functionality	Site Availability
Resolution Case Management - getUserCases	<p>Retrieves a case summary for all cases in which the user is involved. This call retrieves case summary data for the following types of cases:</p> <ul style="list-style-type: none"> ■ Item Not Received cases filed in the Resolution Center. The enumeration value for this case type is EBP_INR. ■ (Item) Significantly Not As Described cases filed in the Resolution Center. The enumeration value for this case type is EBP_SNAD. ■ Item Not Received disputes filed through an eBay legacy dispute system. The enumeration value for this case type is INR. ■ (Item) Significantly Not As Described disputes filed through an eBay legacy dispute system. The enumeration value for this case type is SNAD. ■ Unpaid Item (UPI) disputes created through an eBay legacy dispute system, through the PayPal system, or through the AddDispute call of the Trading API. The enumeration value for this case type is UPI. ■ Cancelled order line items (transactions). Order line items can be cancelled through an eBay site or through the Trading API. The enumeration value for this case type is CANCEL_TRANSACTION. ■ Easy Return items. The enumeration value for this case type is RETURN. ■ Item Not Received cases filed through the PayPal dispute system. The enumeration value for this case type is PAYPAL_INR. ■ (Item) Significantly Not As Described cases filed through the PayPal dispute system. The enumeration value for this case type is PAYPAL_SNAD. 	US, UK, DE
Resolution Case Management - getEBPCaseDetail	<p>Retrieve details and current status on a specific case in which the user is involved. This call retrieves details on the following case types:</p> <ul style="list-style-type: none"> ■ Item Not Received cases filed by the buyer in the Resolution Center. The enumeration value for this case type is EBP_INR. ■ (Item) Significantly Not As Described cases filed by the buyer in the Resolution Center. The enumeration value for this case type is EBP_SNAD. 	US, UK, DE

Table 24-2

API - Call	Functionality	Site Availability
Trading - GetUserDisputes	<p>Retrieves a case summary for all non-eBay Buyer Protection cases in which the user is involved. This call retrieves case summary data for the following types of cases:</p> <ul style="list-style-type: none"> ■ Item Not Received disputes filed through the eBay dispute system. The enumeration value for this case type is INR. ■ (Item) Significantly Not As Described disputes filed through the eBay dispute system. The enumeration value for this case type is SNAD. ■ Unpaid Item (UPI) disputes filed through the eBay dispute system or through the Trading API. The enumeration value for this case type is UPI. ■ Cancelled order line items (transactions) enacted through an eBay site or through the Trading API. The enumeration value for this case type is CANCEL_TRANSACTION. ■ Easy Return items. The enumeration value for this case type is RETURN. 	Available on all sites
Trading - GetDispute	<p>Retrieve details and current status on a specific case in which the user is involved. This call retrieves details on the following case types:</p> <ul style="list-style-type: none"> ■ Item Not Received disputes filed through an eBay legacy dispute system. The enumeration value for this case type is INR. ■ (Item) Significantly Not As Described disputes filed through an eBay legacy dispute system. The enumeration value for this case type is SNAD. ■ Unpaid Item (UPI) disputes created through an eBay legacy dispute system, through the PayPal system, or through the AddDispute call of the Trading API. The enumeration value for this case type is UPI. ■ Cancelled order line items (transactions). Order line items can be cancelled through an eBay site or through the Trading API. The enumeration value for this case type is CANCEL_TRANSACTION. ■ Easy Return items. The enumeration value for this case type is RETURN. 	Available on all sites

For information on tracking and managing disputes through the Trading API dispute calls, see “[Responding to and Tracking Buyer Disputes Through Trading API](#)” on page 477.

Using Filters With the `getUserCases` Call

With the [getUserCases](#) call of the Resolution Case Management API, users have the option of filtering by case status, case type, or date range. Users can also input an eBay **ItemID** (and optionally, a **TransactionID**) to return cases related to a specific item listing or order line item (transaction).

See the [getUserCases](#) call reference for more information on the [getUserCases](#) call.

Item Filter

By using the `itemFilter` field, the user can find user cases associated with a particular item listing by passing in the eBay Item ID. In fact, the user can even "drill-down" to a specific order line item (transaction) by passing in the `transactionId` as well. If the `transactionId` field is used, the user case associated with a specific order line item (transaction) is returned.

Date Range Filter

By using the `creationDateRangeFilter` field, the user can return user cases with creation dates within a specified date range. The `fromDate` and `toDate` must be dates in the past. The `fromDate` can be set back to 18 months in the past. The maximum period of time (or time span) that can be specified using the `fromDate` and `toDate` fields is 90 days. So, if a user wanted to return all user cases from the last 18 months, it would require six separate calls that specified different 90-day intervals. If the `toDate` field is omitted, all cases created from the `fromDate` up to the present (system date) are returned. However, if the time span between the `fromDate` and the system date is longer than 90 days, all cases created starting from the `fromDate` to 90 days after the `fromDate` are returned.

Case Type Filter

By using the `caseTypeFilter` field, the user can find user cases of a specific case type. For example, a user may only want to return PayPal Item Not Received or Significantly Not As Described cases, or maybe they want to return only eBay Buyer Protection Item Not Received or Significantly Not As Described cases. Perhaps a seller knows that a Significantly Not As Described case is filed against them, but they do not know the system in which the case was filed. In that case, they would want to use `SNAD`, `PAYPAL_SNAD`, and `EBP_SNAD` as case type enumeration values in the `caseTypeFilter` field. The `caseTypeFilter` uses Boolean OR logic, which means all matching case types specified in `caseTypeFilter` field are returned.

Case Status Filter

By using the `caseStatusFilter` field, the user can find user cases in a specific status. For example, a user may only want to return opened cases. Or maybe they only want to return closed cases. Perhaps they want to know which cases require an action on their part. In that case, they would want to use `MY_RESPONSE_DUE` and `MY_PAYMENT_DUE` as case status enumeration values in the `caseStatusFilter` field. This `caseStatusFilter` field uses Boolean OR logic, which means all matching case statuses specified in `caseStatusFilter` field are returned.

Retrieving a List of Next Possible Actions on a Case

With the `getActivityOptions` call, a seller can retrieve a list of the next possible actions that they can make on an eBay Buyer Protection case. The activity options that are returned are dependent on the case type and status of the case. Sellers making a `getActivityOptions` call should pay attention to any `buyerPreference` fields that are returned under an applicable activity option container. A true value in a `buyerPreference` field indicates that the corresponding activity option is the buyer's preference for resolving the case. The buyer's preference is stated when the buyer opens the case in the Resolution Center. Only one activity option returned for a seller in the `getActivityOptions` response can be the buyer's preference.

See the `getActivityOptions` call reference for more information on each activity option, including the next step that the seller can take to perform one of the available activity options.

Requesting the Return of a SNAD Item

With the [offerRefundUponReturn](#) call, a seller is able to notify the buyer that a Significantly Not As Described item must be returned before a full refund is issued. The `caseld` container is required in the call input. Optionally, the seller can also specify additional return instructions and an alternative return address. If the seller specifies an “alternative” address, the buyer should use this address instead of the seller’s default shipping address or the return address on file for the seller. The seller will use [issueFullRefund](#) once the Significantly Not As Described item is received.

See the [offerRefundUponReturn](#) call reference for more information on this call.

Issuing a Full Refund to the Buyer

With the [issueFullRefund](#) call, a seller can issue a full refund to a buyer in order to resolve an open eBay Buyer Protection Item Not Received or Significantly Not As Described case. Only the case ID and case type are required in the request, and the three possible results returned in the `fullRefundStatus` field include **AGREED**, **FAILED**, and **SUCCESS**. **AGREED** is always returned if the refund is not being handled in the PayPal system. Non-PayPal refund transactions cannot be tracked by eBay. If **AGREED** is returned, the `refundDate` field is also returned. The `refundDate` gives the buyer an approximate date of the actual refund transaction. **FAILED** indicates that the full refund to the buyer through PayPal was unsuccessful. This does not indicate that the call request failed. If **FAILED** is returned, it is recommended that the seller make a subsequent call at a later time. **SUCCESS** indicates a successful full refund to the buyer through PayPal.

See the [issueFullRefund](#) call reference for more information on this call.

Offering a Partial Refund to the Buyer

With the [offerPartialRefund](#) call, a seller can propose a partial refund amount to the buyer in order to resolve a Significantly Not As Described case. The `caseld` container and the `amount` field (including the `currencyCode` attribute) are required in the call input. Optionally, the seller can include the `comments` field to provide additional information if necessary or applicable.

See the [offerPartialRefund](#) call reference for more information on this call.

Issuing a Partial Refund to the Buyer

With the [issuePartialRefund](#) call, a seller can issue a partial refund to the buyer in order to resolve a Significantly Not As Described case. The `caseld` container and the `amount` field (including the `currencyCode` attribute) are required in the call input. Optionally, the seller can include the `comments` field to provide additional information if necessary or applicable.

The `amount` value must match the value offered to and accepted by the buyer, and the `currencyCode` value must match the actual currency type used to purchase the item. The payment method must be PayPal.

See the [issuePartialRefund](#) call reference for more information on this call.

Providing Shipping Information to the Buyer

With the [provideShippingInfo](#) call, a seller can provide the shipping carrier and shipped date to the buyer. A buyer may request this information when the seller is sending the buyer the original item, a replacement item, or missing/replacement parts for the item.

See the [provideShippingInfo](#) call reference for more information on this call.

Providing Tracking Information to the Buyer

With the [provideTrackingInfo](#) call, a seller can provide package tracking information to the buyer. A buyer may request this information when the seller is sending the buyer the original item, a replacement item, or missing/replacement parts for the item. Tracking information includes the shipping carrier and the tracking number. The eBay system verifies that the specified tracking number is consistent with numbering pattern of the specified shipping carrier, but eBay cannot verify that the tracking number is accurate. Accuracy of the tracking number is the responsibility of the seller.

See the [provideTrackingInfo](#) call reference for more information on this call.

Providing Return Information to the Buyer

With the [provideReturnInfo](#) call, a seller can provide an alternative shipping address to the buyer. The buyer will return the Significantly Not As Described item to this “alternative” address as opposed to the seller’s default shipping address or the return address on file for the seller. The [caselId](#) container and the [address](#) container are required in the call input. Optionally, the seller can include the [returnMerchandiseAuthorization](#) field to provide a Return Merchandise Authorization number to the buyer (if applicable).

See the [provideReturnInfo](#) call reference for more information on this call.

Providing Refund Information to the Buyer

With the [provideRefundInfo](#) call, a German seller (US and UK sellers cannot use this call) can provide a customized message to the buyer regarding a full or partial refund. The [caselId](#) container and the [refundMessage](#) field are required in the call input.

If a German seller has any supporting documents that will act as proof that a refund was issued to the buyer, the seller can upload one to five documents using the [uploadDocuments](#) call.

See the [provideRefundInfo](#) call reference for more information on this call.

Uploading Proof Documents

With the [uploadDocuments](#) call, a German seller (US and UK sellers cannot use this call) can upload one to five documents that serve as proof that an item was shipped to the buyer or proof that an order was fully or partially refunded. The [caselId](#) container, the [proofType](#) field, and at least one [document](#) container are required in the call input. The name and content of each proof document is provided in separate [document](#) containers, and up to five documents may be uploaded with one call. The maximum file size for one document is 1 MB, and supported file types are JPG, GIF, and PNG.

The **proofType** field is used to specify the type of proof documents being uploaded — either proof of shipping or proof of refund.

See the [uploadDocuments](#) call reference for more information on this call.

Offering Other Solution

With the [offerOtherSolution](#) call, a seller can offer the buyer a customized alternative solution to possibly settle an open eBay Buyer Protection case. Possible use cases for using this call include issuing a partial refund to the buyer, providing shipping or refund information to the buyer, or requesting that the buyer return a Significantly Not As Described item before a full refund is issued. The solution, or message to the buyer, is passed into the **messageToBuyer** field in the request.

See the [offerOtherSolution](#) call reference for more information on this call.

Escalating a Case

With the [escalateToCustomerSupport](#) call, a seller can escalate an open eBay Buyer Protection case to eBay customer support. The reason for the escalation must be passed into the **escalationReason** container. Escalation reasons for Item Not Received and Significantly Not As Described cases differ slightly. If **OTHER** is used as the escalation reason, it is recommended that the seller use the optional **comments** field in the request to explain and/or justify the escalation reason.

See the [escalateToCustomerSupport](#) call reference for more information on this call.

Appealing a Case

Buyers and sellers have one opportunity to appeal the decision of a closed eBay Buyer Protection Case. The appeal must be made within 45 days of the case closure date. The appeal is made through the Resolution Center and must be accompanied by supporting documentation.

If the buyer wins an appeal, the buyer is refunded and the seller is not required to reimburse eBay. If the seller wins an appeal, the seller is refunded the reimbursement amount with the reimbursement payment method on file with eBay.

Any eBay or PayPal fees associated with the order line item (transaction) are not to be refunded in the case of an appeal. eBay has sole discretion on making a final decision on all appeals.

With the [appealToCustomerSupport](#) call, a seller can use the Resolution Case Management API to appeal a case decision from eBay customer support. In addition to the case ID and the case type, the seller must also pass in the appeal reason in the request. **appealReason** enumeration values include **DISAGREE_WITH_FINAL_DECISION**, **NEW_INFORMATION**, and **OTHER**. If **OTHER** is used as the appeal reason, it is recommended that the seller use the optional **comments** field in the request to explain and/or justify the appeal reason.

See the [appealToCustomerSupport](#) call reference for more information on this call.

eBay Buyer Protection Platform Notifications

Using the [SetNotificationPreferences](#) call of the Trading API, users can subscribe to the following platform notifications related to the activity of eBay Buyer Protection cases:

- **EBPMyResponseDue**: when a response is due from the user
- **EBPMyPaymentDue**: when a payment is due from the user
- **EBPOtherPartyResponseDue**: when a response is due from the other party involved in the case
- **EBPPaymentDone**: when a payment has been made on the case
- **EBPEscalatedCase**: when a case has been escalated by the user or the other party in the case
- **EBPAppealedCase**: when a case has been appealed by the user or the other party in the case
- **EBPClosedAppeal**: when an appeal on the case has been closed
- **EBPClosedCase**: when a case has been closed
- **EBPOnHoldCase**: when a case has been put on hold

For more information on eBay Buyer Protection Platform Notifications, see “[eBay Buyer Protection Notifications](#)” on page 666.

Chapter 25

Unpaid Item Disputes

In this section:

- ["Creating Unpaid Item Disputes in the Resolution Center" on page 456](#)
- ["Canceling a Transaction in the Resolution Center" on page 456](#)
- ["Creating and Managing Disputes With Trading API" on page 457](#)
- ["Unpaid Item Disputes Process" on page 458](#)
- ["Unpaid Item Dispute States and Consequences" on page 463](#)
- ["Unpaid Item Assistant" on page 463](#)
- ["Communicating with the Buyer" on page 466](#)
- ["Closing an Unpaid Item Dispute" on page 467](#)
- ["Getting Details About a Dispute" on page 469](#)
- ["Reversing a Closed Unpaid Item Dispute" on page 470](#)
- ["Testing Unpaid Item Disputes in the Sandbox" on page 471](#)
- ["Notifications Related to the Unpaid Item Process" on page 473](#)
- ["DisputeState and DisputeActivity for Unpaid Items" on page 473](#)
- ["Implications of Various Values of DisputeActivity" on page 474](#)

A seller can open an Unpaid Item dispute under certain conditions. The seller may hope for the buyer to pay, relist the item, receive a credit for the Final Value Fee, or other positive outcome. Communication between a buyer and seller to resolve an issue is always encouraged by eBay, and is often the fastest and simplest way to resolve a problematic transaction. For tips on working to resolve an issue with a buyer, see ["Resolving Selling Issues" on page 441](#).

Issues that may trigger a seller to open an Unpaid Item dispute:

- Payment issues:
 - The buyer does not pay during the first two days after the order line item (transaction) was created.
 - The buyer refuses to pay.
 - The buyer's check does not clear or the credit card transaction is declined.
 - The buyer wants to use a payment method that the seller doesn't accept.
 - The seller does not receive payment for some other reason.
- Miscellaneous issues:
 - The buyer is no longer a registered eBay user.
 - The buyer wants shipment to an unconfirmed address.
 - The buyer and seller are unable to resolve a disagreement over terms.
 - The buyer returned the item for a refund.
 - The buyer made a mistake and no longer wants the item.

Creating Unpaid Item Disputes in the Resolution Center

On the US, UK, and DE eBay sites, sellers can create, respond to, and settle disputed eBay order line items (transactions) in the Resolution Center. The dispute console for Unpaid Items on other eBay sites vary. Below are a few timelines for the flow of an Unpaid Item dispute in the Resolution Center.

- The seller can open an Unpaid Item case as early as two days or up to 32 days after the order line item (transaction) date.
- eBay notifies the buyer immediately after the Unpaid Item case is opened by the seller.
- The buyer has four days to respond and/or pay for the item after being notified.
- If the buyer does not respond within the four days, the seller is eligible to close the case and receive a final value fee credit. After the case is closed and the final value fee credit is received, the seller is free to extend a Second Chance Offer to another bidder (for an auction listing) or relist the item. The seller may also qualify for [Insertion fee credit](#).

For more information about Unpaid Item disputes, see the [Sellers' Frequently Asked Questions about the eBay Buyer Protection Policy](#) page.

Cancelling a Transaction in the Resolution Center

If a seller cannot go through with a sale, and an agreement is reached with the buyer, the seller may cancel the transaction in the Resolution Center. Possible reasons for canceling a transaction include a lost or broken item, or the item was mistakenly listed by the seller.

Below is the flow for canceling a transaction:

- 1 The seller communicates with the buyer. To cancel the transaction and receive a final value fee credit, a mutual agreement between the buyer and seller is required. See tips on how to [communicate with your buyer](#).
- 2 After getting consent from the buyer to cancel the transaction, the seller opens up a Cancel Transaction case in the Resolution Center. This action notifies the buyer, starts an online dialogue between the seller and buyer, and sets up the case to be tracked in the Resolution Center. The seller can open a Cancel Transaction case up to 45 days after the sale, but generally it is a good practice to open a case immediately.
- 3 The buyer has seven days to respond to the Cancel Transaction case. If the seller has not heard back from the buyer after this time, the seller is eligible to close the case and receive a final value fee credit. The case must be closed within 60 days of the transaction date. On the 61st day, eBay will close the case automatically and the seller will not receive a final value fee credit.

For more information about canceling a transaction in the Resolution Center, see the [Canceling a transaction](#) help page.

Creating and Managing Disputes With Trading API

The following table summarizes the key activities related to seller disputes and the dispute calls of the Trading API.

Table 25-1

Actions for buyers and sellers	API option
<i>Seller:</i> create a new Unpaid Item dispute or cancel a transaction	AddDispute . See “ Unpaid Item Dispute States and Consequences ” on page 463.
<i>Buyer:</i> respond to a dispute opened by a seller.	None. A buyer must use the Resolution Center or other dispute console to respond to a dispute.
<i>Seller:</i> comment on an open dispute	AddDisputeResponse . See “ Communicating with the Buyer ” on page 466.
<i>Seller:</i> close a dispute under the following circumstances: <ul style="list-style-type: none"> ■ - buyer and seller completed the transaction ■ - buyer and seller agreed to not complete the transaction ■ - seller wants to end communication with the buyer 	AddDisputeResponse . See “ Closing an Unpaid Item Dispute ” on page 467.
<i>Any user:</i> get details about a <i>specific</i> dispute in which the user was involved. This call is not applicable to eBay Buyer Protection Item Not Received or Significantly Not As Described cases.	GetDispute . See “ Getting Details About a Dispute ” on page 469.
<i>Any user:</i> get a summarized account of <i>all</i> disputes in which the user was involved. This call is not applicable to eBay Buyer Protection Item Not Received or Significantly Not As Described cases.	 GetUserDisputes . See “ Getting Details About a Dispute ” on page 469.
<i>Seller:</i> change (reverse) a closed dispute from unresolved to resolved	SellerReverseDispute . See “ Reversing a Closed Unpaid Item Dispute ” on page 470.

Using AddDispute

The [AddDispute](#) creates an Unpaid Item dispute for, or cancels one order line item (transaction). In addition to the authorization token, the [AddDispute](#) requires either a TransactionID/ItemID pair or the OrderLineItemID value. It also requires the reason and explanation for the dispute.

Only `BuyerHasNotPaid` and `TransactionMutuallyCanceled` are applicable values for the `DisputeReason` in the [AddDispute](#).

The `DisputeExplanation` field can be thought of as the “sub-reason” for the dispute. Consequently, only certain values of `DisputeExplanation` are applicable based on the specified `DisputeReason`. For the latest information on which `DisputeExplanation` values apply to each value of `DisputeReason`, see the call reference documentation. Alternatively, see the following type definitions:

[DisputeReasonCodeType](#)

[DisputeExplanationCodeType](#)

Here is a snapshot of the applicable values for these fields:

Table 25-2 Valid DisputeExplanations per DisputeReason

DisputeReason	Allowed DisputeExplanations
BuyerHasNotPaid	BuyerHasNotResponded, BuyerNoLongerRegistered, BuyerNotClearedToPay, BuyerRefusedToPay, ShippingAddressNotConfirmed, OtherExplanation, BuyerNotPaid, BuyerPaymentNotReceivedOrCleared, SellerDoesntShipToCountry Deprecated: PaymentMethodNotSupported, ShipCountryNotSupported.
TransactionMutuallyCanceled	BuyerNoLongerWantsItem, BuyerPurchasingMistake, ShippingAddressNotConfirmed, BuyerReturnedItemForRefund, UnableToResolveTerms, OtherExplanation Deprecated: PaymentMethodNotSupported

A call to [AddDispute](#) returns a dispute ID. The **DisputeID** is useful for several Trading API dispute calls.

An error results if you call [AddDispute](#) in these cases:

- It has been fewer than four full days or greater than 32 days since the order line item (transaction) was created (and the buyer is still a registered eBay user).
- The buyer paid via PayPal (in which case there should be no issue or dispute).
- The order line item (transaction) is already the subject of another dispute. There can only be one dispute per order line item (transaction).

See “[AddDispute Sample](#)” on page 773 and “[Testing AddDispute](#)” on page 772.

Notifications

Creating an Unpaid Item dispute triggers a [UPISellerOpenedDispute](#) notification. The notification returns a [Dispute](#) element. To subscribe to this notification and receive messages by email or posted to a URL, use [SetNotificationPreferences](#) (see “[SetNotificationPreferences](#)” on page 929).

Unpaid Item Disputes Process

In October 2009, changes were introduced in the Unpaid Item Disputes process around a **DisputeReason** value of [BuyerHasNotPaid](#).

It is the request version that determines which process applies: If you created a UPI dispute with a request version lower than 637, the older process applies and the various platform notifications for messaging are still supported. If the request version is 637 or greater when you create a UPI dispute, the newer process applies, and several platform notifications no longer apply. (The reason the notifications are not needed is because messages between buyer and seller are no longer added to the dispute.)

Compare the left sides of the “before and after” diagrams, “[Unpaid Item Dispute Diagram for Version 637 or Later](#)” on page 460 and “[Unpaid Item Dispute Diagram for Versions Earlier Than 637](#)” on page 461.

Table 25-3 Changes Related to BuyerHasNotPaid Dispute Reason

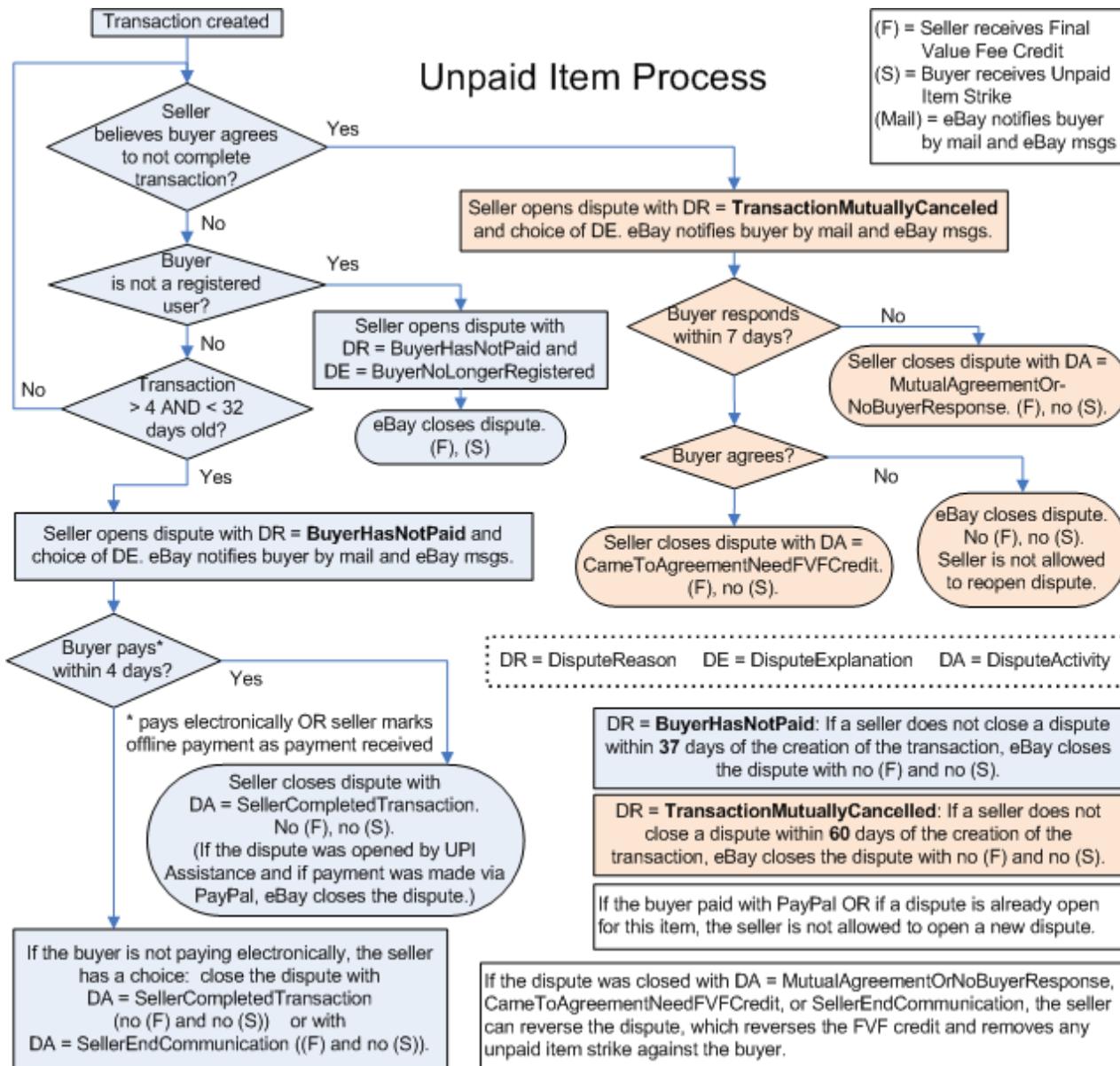
Older Process	Newer Process
The soonest a seller can open an Unpaid Item dispute is seven days from the creation of the order line item (transaction).	The soonest a seller can open an Unpaid Item dispute is two days from the creation of the order line item (transaction).
The latest a seller can open an unpaid item dispute is 45 days from the creation of the order line item (transaction).	The latest a seller can open an unpaid item dispute is 32 days from the creation of the order line item (transaction).
The soonest that eBay or the seller can act on a dispute when the buyer has not responded is seven days from when the dispute was opened.	The soonest that eBay or the seller can act on a dispute when the buyer has not paid is four days from when the dispute was opened.
If the buyer has not responded in some way within seven days, the seller can close the dispute with MutualAgreementOrNoBuyerResponse .	MutualAgreementOrNoBuyerResponse is no longer valid in the BuyerHasNotPaid flow.
If an Unpaid Item dispute is still open 60 days after the order line item (transaction) was created, eBay closes it automatically, regardless of what actions the buyer or seller might have taken.	If an Unpaid Item dispute is still open 37 days after the order line item (transaction) was created, eBay closes it automatically, regardless of what actions the buyer or seller might have taken. eBay emails a reminder to the seller on day 36. (No change for MutualAgreementOrNoBuyerResponse : it is still 60 days.)
If the buyer has paid with PayPal, eBay closes the dispute with DA = SellerCompletedTransaction.	If the buyer has paid (whether electronically or the seller has marked as payment received for an offline payment), the seller closes the dispute with DA = SellerCompletedTransaction. If the dispute was opened by the Unpaid Item Assistant and the buyer paid via PayPal, eBay closes the dispute.
—	There are three more DisputeStates : UnpaidItemOpen , UPIAssistanceDisabledByBay , UPIAssistanceDisabledBySeller . For each of these, the valid DisputeActivities are SellerCompletedTransaction and SellerEndCommunication . When payment has not been received, SellerEndCommunication can be used four days after the creation of the BuyerHasNotPaid dispute.
—	There are five more DisputeExplanations : BuyerNotPaid , BuyerPaymentNotReceivedOrCleared , SellerDoesntShipToCountry , UPIAssistance , UPIAssistanceDisabled .
—	The Unpaid Item Assistant feature automates the process of opening and closing BuyerHasNotPaid disputes. See “ Unpaid Item Assistant ” on page 463.
A seller can use AddDisputeResponse to comment on a dispute.	Comments are no longer passed in or stored in an Unpaid Item dispute; a seller can no longer add comments to a dispute with AddDisputeResponse (buyer and seller are to communicate via other normal means, outside of the dispute record itself).

Table 25-3 Changes Related to BuyerHasNotPaid Dispute Reason

Older Process	Newer Process
Platform notifications are triggered upon creating a dispute or adding a comment to a dispute.	These platform notifications are not triggered by AddDispute or AddDisputeResponse for Unpaid Item disputes : BuyerResponseDispute , SellerClosedDispute , SellerOpenedDispute , SellerRespondedToDispute . (Note that for the case of mutual agreement to cancel the order line item (transaction), the applicable notifications still apply.)

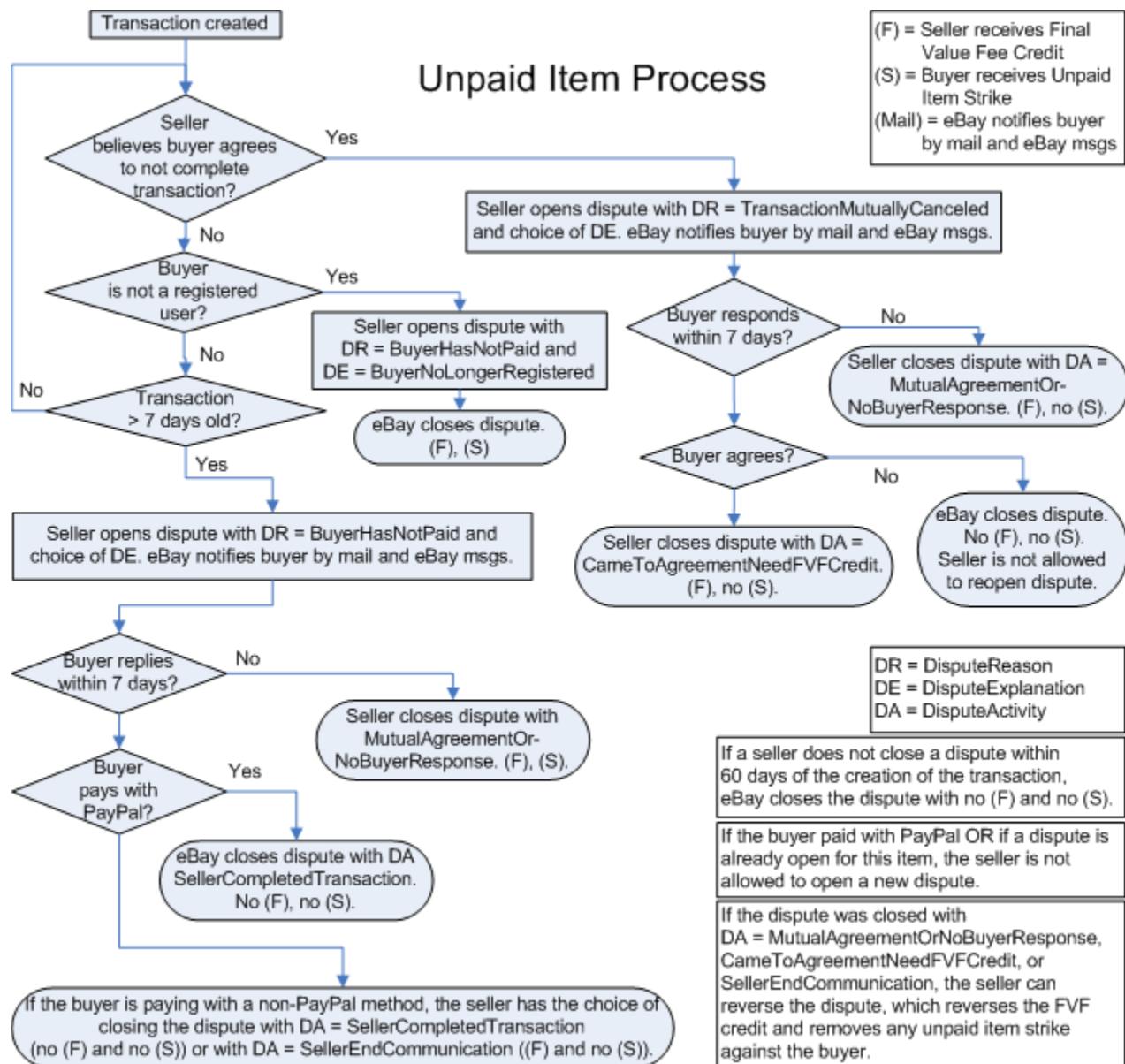
Unpaid Item Dispute Diagram for Version 637 or Later

The following diagram represents the Unpaid Item dispute lifecycle for request versions of 637 or later.



Unpaid Item Dispute Diagram for Versions Earlier Than 637

The following diagram represents the Unpaid Item dispute lifecycle for request versions earlier than 637.



Unpaid Item Dispute States and Consequences

The following table presents various states and the corresponding consequences of opening a dispute for a given order line item (transaction):

Table 25-4 Implications of Opening a Dispute

State when the dispute is opened	Implications of opening the dispute
The buyer is no longer a registered eBay user.	<ul style="list-style-type: none"> ■ A credit for the Final Value Fee (FVF) is issued to the seller immediately. ■ eBay does not notify the buyer. ■ The seller can immediately close the dispute.
The buyer has not paid.	<ul style="list-style-type: none"> ■ eBay sends the buyer an Unpaid Item Reminder. ■ To avoid an Unpaid Item Strike, the buyer must respond within four full days. ■ The seller cannot close the dispute until the buyer has responded or four full days have passed, whichever comes first.
The seller wishes to claim that the buyer agrees to not complete the order line item (transaction).	<ul style="list-style-type: none"> ■ The seller must wait for buyer agreement or for seven full days to pass with no buyer response to close the dispute. ■ If the buyer responds in agreement, the seller can close the dispute immediately. ■ If the buyer responds in disagreement, eBay closes the dispute and the seller is not allowed to reopen a dispute for that order line item (transaction).

Unpaid Item Assistant

In this section:

[“Introduction” on page 463](#)

[“Unpaid Item Assistant Preferences” on page 464](#)

[“API Support for Unpaid Item Assistant” on page 464](#)

Introduction

The Unpaid Item Assistant feature allows eBay to open and close Unpaid Item disputes automatically on a seller's behalf. There is both Web and API support for this mechanism.

Unpaid Item Assistant

<http://pages.ebay.com/help/sell/unpaid-item-assistant.html>

Understanding the Unpaid Item Process

<http://pages.ebay.com/sell/UPI/standardprocess/index.html>

If the seller would like to automatically relist an item after an Unpaid Item dispute is resolved (without payment) through the Unpaid Item Assistant, the `AutoRelist` field in `SetUserPreferences` should be set to true.

Unpaid Item Assistant Preferences

To set up the Unpaid Item Assistant to automatically open Unpaid Item cases, the seller must opt into the Unpaid Item Assistant feature through the Unpaid Item Assistant Preferences in My eBay or through the [SetUserPreferences](#) call. With the Unpaid Item Assistant Preferences, the seller can control the following actions:

- How many days to wait (without payment from buyer) before the Unpaid Item Assistant opens an Unpaid Item case
- eBay automatically relisting the item when a Unpaid Item case is closed with no payment from buyer
- eBay automatically requesting a Giving Works donation refund for the seller when an Unpaid Item case is closed with no payment from buyer
- Creating an “Exclusion List” of buyers. The Unpaid Item Assistant will not automatically create Unpaid Item cases for these “excluded” buyers.
- eBay automatically sending an email to the seller when an Unpaid Item case is opened (this option has no API support)
- eBay automatically sending an email to the seller when an Unpaid Item case is closed (this option has no API support)

API Support for Unpaid Item Assistant

In this section:

[“Survey of Types” on page 464](#)

[“Turning Off Automatic Dispute Handling for an Item” on page 465](#)

[“Automatic Relisting of Items” on page 465](#)

Survey of Types

The following three enumeration values in [DisputeStateCodeType](#) are related to the Unpaid Item Assistant feature. [DisputeStateCodeType](#) is used by the [GetDispute](#) and [GetUserDisputes](#) calls of the Trading API.

- [UnpaidItemOpen](#): indicates an open Unpaid Item case
- [UPIAssistanceDisabledByeBay](#): the Unpaid Item dispute created by the Unpaid Item Assistant has been canceled by eBay. eBay will take this action if it detects that a payment has been initiated by the buyer.
- [UPIAssistanceDisabledBySeller](#): the Unpaid Item dispute created by the Unpaid Item Assistant has been canceled by the seller. The seller might take this action if the seller has communicated with the buyer and the seller wishes to extend the time allowed for the payment of the item.

The following two values of [DisputeExplanationCodeType](#), set by eBay, signify that the Unpaid Item Assistant was or is being used for the dispute: [UPIAssistance](#), [UPIAssistanceDisabled](#).

[SetUserPreferences](#) and [GetUserPreferences](#) have a [UnpaidItemAssistancePreferences](#) container. This container is defined by the [UnpaidItemAssistancePreferencesType](#). One of the fields in the container is [OptInStatus](#). A value of ‘true’ in this field indicates that the seller has opted into the Unpaid Item Assistant feature and wants eBay to automatically file disputes when a buyer has not paid when the [UnpaidItemAssistancePreferences.DelayBeforeOpeningDispute](#) value is reached. See the Call Reference for [UnpaidItemAssistancePreferencesType](#) to learn more about its elements.

Turning Off Automatic Dispute Handling for an Item

eBay recommends that developers and certified providers monitor the [OptInStatus](#) and [AutoRelist](#) status for each seller, depending on the Unpaid Item Assistant features you offer.

After a Unpaid Item dispute is created with the Unpaid Item Assistant, the buyer has four days to pay for the item before the Unpaid Item Assistant automatically closes the case, gives the seller a final value fee credit, and records an Unpaid Item strike on the buyer’s account. (See exceptions below, where an automatic dispute is converted to a manual dispute.)

If the seller wants to intervene and extend the time for the buyer to pay (e.g. the buyer contacts the seller and makes arrangement for later payment), the seller calls [DisableUnpaidItemAssistance](#) with the [DisputeID](#), the [ItemID/TransactionID](#) pair, or [OrderLineItemID](#) for that item.

The following are reasons why eBay may convert an automatically handled dispute to a manually handled dispute:

- The buyer has already partially paid for the order line item (transaction).
- The buyer was refunded for the item.
- Payment was marked as pending.
- The listing was ended by eBay customer support.
- The seller marked the item as unpaid.

Automatic Relisting of Items

If Unpaid Item disputes are being created and handled automatically with Unpaid Item Assistant, the seller can choose whether the items associated with the disputes are automatically relisted by eBay. The value of [UnpaidItemAssistancePreferencesType.AutoRelist](#) reflects the seller’s preference for automatically relisting items after Unpaid Item cases are closed. For a multi-quantity, fixed-price listing, eBay merely adjusts the quantity if [AutoRelist](#) is set to true. This flag is set with [SetUserPreferences](#).

Avoiding Automated Dispute Handling for Specific Buyers

A seller may wish to disable automatic dispute handling for specific buyers. The seller specifies these “excluded” buyers by using the [UnpaidItemAssistancePreferences.ExcludedUser](#) field in [SetUserPreferences](#).

For example, imagine there is a buyer who reliably pays the seller for items within 10 days. The seller has complete faith in this buyer. The seller does not want the automated dispute process to automatically treat this transaction as an “unpaid item” and file a dispute. So, the seller includes this buyer’s user id in the [ExcludedUser](#) field. A separate [ExcludedUser](#) field is required for each user, and a maximum of 5,000 users may be specified.

To see the current list of buyers that the seller is excluding from the automated dispute process, call [GetUserPreferences](#) with [ShowUnpaidItemAssistanceExclusionList](#) set to true.

To completely clear the list of buyers, call [SetUserPreferences](#) with [UnpaidItemAssistancePreferencesType.RemoveAllExcludedUsers](#) set to true.

Requesting a Donation Refund for Charity Listings

Similar to the automatic relisting of items, a seller can also set the [UnpaidItemAssistancePreferencesType.AutoOptDonationRefund](#) preference that will trigger eBay to automatically request a donation refund after UPI Assistant cases are closed with no payment. This feature is only applicable to charity listings.

eBay routes the donation refund request to Mission Fish, and then Mission Fish credits the seller's account. This flag is set with [SetUserPreferences](#).

Communicating with the Buyer

Once a dispute is created, eBay contacts the buyer by email and pop-up messages, encouraging the buyer to pay. In general, eBay encourages the buyer and seller to communicate in an attempt to resolve the situation.

Version 637 or Greater

With a request version of 637 or greater, a seller can no longer add a comment or response to a dispute. A seller and buyer would communicate by other normal means, such as emails or eBay messages.

Prior to Version 637

With a request version earlier than 637, a seller can add a comment or response to a dispute with [AddDisputeResponse](#) at any time, except when the state of the dispute is [Locked](#), [Closed](#) or [PendingResolve](#). (See " [DisputeState and DisputeActivity for Unpaid Items](#)" on page 473.) The seller can add comments up to 60 days after the dispute was created. Adding a comment does not close the dispute, generate an Unpaid Item Strike for the buyer, or credit the Final Value Fee to the seller. The seller is limited to 25 comments or responses.

To add a comment via API, call [AddDisputeResponse](#) on behalf of the seller, providing the ID of the dispute, setting [DisputeActivity](#) to [SellerAddInformation](#), and sending [MessageText](#). ([DisputeID](#) is returned by [AddDispute](#) when opening a dispute. You can also retrieve the IDs of disputes open for a user by calling [GetUserDisputes](#)—see " [Getting Details About a Dispute](#)" on page 469).

Adding a comment via [AddDisputeResponse](#) triggers a " [SellerRespondedToDispute](#)" on page 760 notification. Subscribe to the notification with [SetNotificationPreferencesCall](#). See also " [Notifications Related to the Unpaid Item Process](#)" on page 473.

Closing an Unpaid Item Dispute

In this section:

- ["If the Buyer and Seller Complete the Transaction" on page 468](#)
- ["If the Seller Chooses to Not Complete the Transaction" on page 468](#)
- ["If the Seller Decides to End Communication with the Buyer" on page 468](#)
- ["Notifications" on page 469](#)

A seller can close a dispute and receive a final value fee credit under the following conditions:

- The buyer has responded or paid
- The **DisputeReason** is **BuyerHasNotPaid** and the dispute has been open for more than four days
- The **DisputeReason** is **MutualAgreementOrNoBuyerResponse** and the dispute has been open for more than seven days

If an Unpaid Item dispute is still open 37 days (for **BuyerHasNotPaid**) or 60 days (for **MutualAgreementOrNoBuyerResponse**) after the order line item (transaction) was created, eBay closes it automatically, regardless of what actions the buyer or seller might have taken. When the state is **BuyerHasNotPaid**, eBay emails a reminder to the seller on day 36.

To close a dispute, the seller uses the Resolution Center, the dispute console (if applicable), or the **AddDisputeResponse**. Note that a seller can close a dispute with **AddDisputeResponse** only if that dispute was created manually and not created by the Unpaid Item Assistant (see "["Unpaid Item Assistant" on page 463](#)).

To close a dispute with **AddDisputeResponse**, provide the value of **DisputeActivity** that indicates the reason for closing. The current **DisputeState** for the dispute determines which values of **DisputeActivity** are allowed. The **DisputeState** for a case can be retrieved with the **GetDispute** call. See "["DisputeState and DisputeActivity for Unpaid Items" on page 473](#)".

A dispute cannot be closed if the **DisputeState** value is **PendingResolve**.

The choice of **SellerEndCommunication** for **DisputeActivity** is only allowed if the **DisputeState** is **BuyerFirstResponsePayOptionLateResponse** or **BuyerFirstResponseNoPayOptionLateResponse**.

Note that if there is any kind of buyer response, the step requesting that the dispute be closed must not be automated. eBay prohibits automation of this step to avoid false claims.

Depending on the choice of **DisputeActivity**, the buyer may or may not receive an Unpaid Item Strike, and the seller may or may not receive a Final Value Fee credit. See "["Implications of Various Values of DisputeActivity" on page 474](#)".

The seller is eligible for relist credit if:

- the buyer does not pay for the item
- the seller selected **SellerEndCommunication** as the activity
- the seller selected **MutualAgreementOrNoBuyerResponse** as the activity

For details on relist credit, see the following on the eBay Web site:

<http://pages.ebay.com/help/sell/questions/free-relist.html>

See also “[Testing Unpaid Item Disputes in the Sandbox](#)” on page 471.

If the Buyer and Seller Complete the Transaction

If the buyer pays, the seller should close the dispute with a **DisputeActivity** of **SellerCompletedTransaction**. The seller receives payment, so there is no Final Value Fee credit for the seller and no Unpaid Item Strike against the buyer.

If the Seller Chooses to Not Complete the Transaction

The seller can declare (on Web or using **AddDisputeResponse**) that an agreement has been reached with the buyer to not complete the order line item (transaction) – whether or not the seller and buyer had such a discussion or agreement. To declare this via **AddDisputeResponse**, set **DisputeReason** to **TransactionMutuallyCanceled** and set **DisputeExplanation** to the reason the seller believes the buyer agrees, such as **BuyerNoLongerWantsItem** or **BuyerReturnedItemForRefund**, etc.

After making such a declaration, the seller waits for the buyer to respond via the Resolution Center or dispute console or waits seven full days, whichever comes first, before the seller can close the dispute.

If the buyer agrees that they will not complete the order line item (transaction), the seller can close the dispute with a **DisputeActivity** of **CameToAgreementNeedFVFCredit**.

If the buyer responds in agreement, or if seven full days pass with no response from the buyer, you can enable the seller to close the dispute with a **DisputeActivity** of **MutualAgreementOrNoBuyerResponse**.

If the Seller Decides to End Communication with the Buyer

If the buyer does not respond or if the seller finds the communication to not be effective, the seller can close the dispute with a **DisputeActivity** of **SellerEndCommunication** or **MutualAgreementOrNoBuyerResponse**.

The seller receives a Final Value Fee credit for the order line item (transaction) that was never completed, and the buyer receives an Unpaid Item Strike against his account.

Notifications

Calling [AddDisputeResponse](#) to close a dispute triggers a “[SellerClosedDispute](#)” on page 754 notification. Subscribe to the notification with [SetNotificationPreferencesCall](#). See also “[Notifications Related to the Unpaid Item Process](#)” on page 473.

Getting Details About a Dispute

In this section:

- “[Overview](#)” on page 469
- “[GetDispute](#)” on page 469
- “ [GetUserDisputes](#)” on page 470

Overview

Either buyer or seller can retrieve information on any dispute in which he or she has been involved, for up to five years after the dispute was closed.

The [GetUserDisputes](#) returns details on one or multiple disputes. The [GetDispute](#) return details about a single dispute, a dispute which is specified through the [DisputeID](#) field in the request. If the buyer or seller knows the ID for a specific dispute, he or she can also use the Resolution Center or dispute console to gather information for that specific dispute.

Summary information and the ID for Unpaid Item disputes can also be retrieved with the [getUserCases](#) call of the Resolution Case Management API. In the [getUserCases](#) response, the ID for the Unpaid Item dispute is found in the [cases.caseSummary.caseld.id](#) field. If you are using the [getUserCases](#) call to retrieve Unpaid Item disputes, it is recommended that you use the case type filter, and pass in [UPI](#) in the [caseTypeFilter.caseType](#) field of the [getUserCases](#) request.

GetDispute

If you know the ID for a specific dispute, you can call [GetDispute](#). Dispute details returned include such information as whether eBay sent a notice to the buyer, whether the buyer received an Unpaid Item Strike, and whether the seller was credited the Final Value Fee.

The returned [DisputeState](#) returned is useful when you want to use [AddDisputeResponse](#) to close a dispute. See “[AddDisputeResponse](#)” on page 775.

The dispute has a [DisputeRecordType](#) element that specifies the type of dispute, which can be either [UnpaidItem](#) or [ItemNotReceived](#).

The dispute also has a [DisputeCreditEligibility](#) element. [DisputeCreditEligibility](#) does not state whether a credit *ultimately* will or will not be issued to the seller, but only whether the dispute is *currently eligible* for credit.

In most cases, the seller must wait four or seven full days for the buyer to respond ([BuyerHasNotPaid](#) vs [MutualAgreementOrNoBuyerResponse](#), respectively) after opening the

dispute; thus, the value of **DisputeCreditEligibility** is **InEligible** during that time. Exception: if the buyer is no longer a registered user, the value may be **Eligible** immediately.

See [GetDispute Samples](#) and [Testing GetDispute](#) in the eBay Trading API Call Reference. See also “[Testing Unpaid Item Disputes in the Sandbox](#)” on page 471.

For another sample of a details of a dispute, see “[Testing Unpaid Item Disputes in the Sandbox](#)” on page 471

GetUserDisputes

You can use the Resolution Center or dispute console to retrieve the full list of disputes in which you are involved, or you can call [GetUserDisputes](#). Note that the call might return a large number of disputes, so using the available input filters is typically a good idea. See “[GetUserDisputes](#)” on page 878 for more information.

See [GetUserDisputes Samples](#) and [Testing GetUserDisputes](#) in the eBay Trading API Call Reference.

Reversing a Closed Unpaid Item Dispute

In this section:

- “[Overview](#)” on page 470
- “[SellerReverseDispute](#)” on page 470

Overview

Sometimes, a buyer and seller resolve a dispute after the dispute was previously closed with a status of the seller no longer wishing to communicate (a **DisputeActivity** of **SellerEndCommunication**). For example, a seller might discover that the buyer actually paid for the item, or perhaps the buyer offered to reimburse the seller's fees in return for having the strike removed. Regardless, the seller is interested in receiving a Final Value Fee credit and the buyer is interested in being free of the Unpaid Item Strike.

A seller can use the Resolution Center, dispute console, or use [SellerReverseDispute](#) to signify that “We've completed the transaction and we're both satisfied.” This is referred to as “reversing a closed dispute”. Any Final Value Fee credit the seller received and any Unpaid Item Strike made against the buyer are reversed.

SellerReverseDispute

A dispute can only be reversed by the seller with [SellerReverseDispute](#) if it was closed with a **DisputeActivity** set to **SellerEndCommunication**, **CameToAgreementNeedFVFCredit**, or **MutualAgreementOrNoBuyerResponse**. (See also “[Implications of Various Values of DisputeActivity](#)” on page 474.)

When calling [SellerReverseDispute](#), the **DisputeID** and **DisputeResolutionReason** fields are required. The **DisputeResolutionReason** describes why the closed dispute is being reversed.

When reversing a closed dispute with this call, the only applicable values for **DisputeResolutionReason** are **CameToAgreement**, **BuyerPaidAuctionFees**, **SellerReceivedPayment** and **OtherResolution**.

See also “[SellerReverseDispute Samples](#)” on page 923 and “[Testing SellerReverseDispute](#)” on page 923.

Testing Unpaid Item Disputes in the Sandbox

To prepare to test Unpaid Item disputes in Sandbox, you first use a seller account to list a Buy It Now item with [AddItem](#). Then, using a buyer account, you place an offer on the item using [PlaceOffer](#).

In the real world (eBay Marketplace), a seller must wait the four or seven days before filing a dispute (except for a couple of exceptions). However, in the Sandbox, a seller can immediately file a dispute using [AddDispute](#).

The example below tests one of the more complicated scenarios, that of a seller filing a dispute with the claim that both buyer and seller agree to mutually cancel the order line item (transaction), but then the buyer does not respond to confirm agreement.

- 1 Seller: List a Buy It Now item.
- 2 Buyer: Place an offer on the item to end the item.
- 3 Seller: open a dispute (“[Unpaid Item Dispute States and Consequences](#)” on page 463) with **DisputeReason** and **DisputeExplanation** values of **TransactionMutuallyCanceled** and **BuyerNoLongerWantsItem**, respectively. (See “[If the Seller Chooses to Not Complete the Transaction](#)” on page 468 for insights into **TransactionMutuallyCanceled**.)

Example 25-1 Opening a dispute

```
<?xml version="1.0" encoding="utf-8"?>
<AddDisputeRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <ItemID>9202501086</ItemID>
  <TransactionID>0</TransactionID>
  <DisputeReason>TransactionMutuallyCanceled</DisputeReason>
  <DisputeExplanation>BuyerNoLongerWantsItem</DisputeExplanation>
</AddDisputeRequest>
```

- 4 Since the Sandbox does not eliminate the waiting period, you will need to let the dispute age the four or seven full days without response from the buyer (days depending on **BuyerHasNotPaid** vs **MutualAgreementOrNoBuyerResponse**).
- 5 Check the state of the dispute (see “[Getting Details About a Dispute](#)” on page 469). The state of **MutualWithdrawalAgreementLate** is fitting: the buyer did not respond to the seller’s claim that the seller and buyer mutually agreed to not complete the order line item (transaction). Note that

the values `TransactionMutuallyCanceled` and `BuyerNoLongerWantsItem` are from the original dispute.

Example 25-2 Checking the state of the dispute

```
<GetDisputeResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2006-05-16T17:44:38.675Z</Timestamp>
  <Ack>Success</Ack>
  <Version>459</Version>
  <Build>e459_core_Bundled_2927347_R1</Build>
  <Dispute>
    <DisputeID>262550</DisputeID>
    <DisputeRecordType>UnpaidItem</DisputeRecordType>
    <DisputeState>MutualWithdrawalAgreementLate</DisputeState>
    <DisputeStatus>WaitingForBuyerResponse</DisputeStatus>
    <BuyerUserID>buyer</BuyerUserID>
    <SellerUserID>seller</SellerUserID>
    <TransactionID>0</TransactionID>
    <Item>
      <ItemID>9202501086</ItemID>
      <ListingDetails>
        <StartTime>2006-04-15T17:32:48.000Z</StartTime>
        <EndTime>2006-04-15T17:33:17.000Z</EndTime>
      </ListingDetails>
      <Quantity>1</Quantity>
      <SellingStatus>
        <ConvertedCurrentPrice currencyID="USD">10.0</ConvertedCurrentPrice>
        <CurrentPrice currencyID="USD">10.0</CurrentPrice>
      </SellingStatus>
      <Site>US</Site>
      <Title>Dachshund Salt and Pepper Shakers</Title>
    </Item>
    <DisputeReason>TransactionMutuallyCanceled</DisputeReason>
    <DisputeExplanation>BuyerNoLongerWantsItem</DisputeExplanation>
    <DisputeCreditEligibility>InEligible</DisputeCreditEligibility>
    <DisputeCreatedTime>2006-05-16T17:41:14.000Z</DisputeCreatedTime>
    <DisputeModifiedTime>2006-05-16T17:44:01.000Z</DisputeModifiedTime>
    <DisputeMessage>
      <MessageID>633599</MessageID>
      <MessageSource>eBay</MessageSource>
      <MessageCreationTime>2006-05-16T17:41:20.000Z</MessageCreationTime>
      <MessageText>An Unpaid Item dispute has been opened for the following item:  
NewSchema item  
title (#9202501086) Reason given for Unpaid Item: We have both agreed not to complete the transaction.  
Buyer actions reported by seller: The buyer did not want item.</MessageText>
    </DisputeMessage>
  </Dispute>
</GetDisputeResponse>
```

- 6 Seller:** Close the dispute (see “[Closing an Unpaid Item Dispute](#)” on page 467) with a **DisputeActivity** value of **MutualAgreementOrNoBuyerResponse**.

Example 25-3 Closing the dispute

```
<AddDisputeResponseRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <Version>459</Version>
  <DisputeID>262550</DisputeID>
  <DisputeActivity>MutualAgreementOrNoBuyerResponse</DisputeActivity>
  <MessageText>Will close this dispute.</MessageText>
</AddDisputeResponseRequest>
```

- 7** Check the **DisputeState** and **DisputeStatus** via [GetDispute](#): they should be **Closed** and **ClosedNoFVFCreditNoStrike**, respectively.

Notifications Related to the Unpaid Item Process

A developer or an application can subscribe to notifications that are sent when various events occur on Unpaid Item disputes. To subscribe to a notification, use [SetNotificationPreferences](#).

Table 25-5 Notifications related to disputes

Event	Notification
A seller opens a dispute, on the eBay site or with AddDispute .	“ SellerOpenedDispute ” on page 756
A buyer responds to a dispute on the eBay site.	“ BuyerResponseDispute ” on page 751
A seller responds to a dispute, on the eBay site or with AddDisputeResponse .	“ SellerRespondedToDispute ” on page 760
A seller closes a dispute, on the eBay site or with AddDisputeResponse .	“ SellerClosedDispute ” on page 754

DisputeState and DisputeActivity for Unpaid Items

DisputeActivity is a required field in the [AddDisputeResponse](#) request used to specify the activity being made on the dispute. Acceptable values of **DisputeActivity** depend on the value of **DisputeState**. (To get the **DisputeState** for a dispute, call [GetDispute](#) with the **DisputeID**. See “[Getting Details About a Dispute](#)” on page 469.)

See also the Type documentation for the following:

[DisputeStateCodeType](#)

[DisputeActivityCodeType](#)

Table 25-6 DisputeActivity and DisputeState for Unpaid Item Disputes

When DisputeState Is	DisputeActivity Can Be
BuyerFirstResponseNoPayOption	SellerAddInformation SellerCompletedTransaction
BuyerFirstResponseNoPayOptionLateResponse	SellerAddInformation SellerCompletedTransaction SellerEndCommunication
BuyerFirstResponsePayOption	SellerAddInformation SellerCompletedTransaction
BuyerFirstResponsePayOptionLateResponse	SellerAddInformation SellerCompletedTransaction SellerEndCommunication
Closed	None
Locked	None
MutualAgreementOrBuyerReturningItem	SellerAddInformation
MutualCommunicationNoPayOption	SellerAddInformation SellerCompletedTransaction SellerEndCommunication CameToAgreementNeedFVFCredit MutualAgreementOrNoBuyerResponse
MutualCommunicationPayOption	SellerAddInformation SellerCompletedTransaction SellerEndCommunication CameToAgreementNeedFVFCredit MutualAgreementOrNoBuyerResponse
MutualWithdrawalAgreement	SellerAddInformation MutualAgreementOrNoBuyerResponse
MutualWithdrawalAgreementLate	SellerAddInformation SellerCompletedTransaction SellerEndCommunication MutualAgreementOrNoBuyerResponse
PendingResolve	None
Reserved	SellerAddInformation SellerCompletedTransaction
UnpaidItemOpen	SellerCompletedTransaction SellerEndCommunication
UPIAssistanceDisabledByeBay	SellerCompletedTransaction SellerEndCommunication
UPIAssistanceDisabledBySeller	SellerCompletedTransaction SellerEndCommunication

Implications of Various Values of DisputeActivity

Table 25-7 shows the applicable **DisputeActivity** values for Unpaid Item disputes and the ending result of the activity if the **AddDisputeResponse** call is successful.

Table 25-7 Results of DisputeActivity

DisputeActivity Value	Result
SellerAddInformation. Seller wants to add information or respond to an email from the buyer. When this value is chosen, use MessageText to pass a message.	<ul style="list-style-type: none"> The dispute remains open.
SellerCompletedTransaction. The buyer and seller have completed the order line item (transaction) and are both satisfied.	<ul style="list-style-type: none"> The dispute is closed. The seller does not receive a Final Value Fee credit. The buyer does not receive a strike.
CameToAgreementNeedFVFCredit. The seller indicates that buyer and seller have agreed not to complete the order line item (transaction), but the buyer must confirm. eBay generates an email to the buyer.	<p>If the buyer confirms:</p> <ul style="list-style-type: none"> The dispute is closed. The buyer does not receive a strike. The seller receives a Final Value Fee credit. <p>If the buyer disagrees:</p> <ul style="list-style-type: none"> The dispute is closed. The buyer does not receive a strike. The seller does not receive a Final Value Fee credit.
SellerEndCommunication. Seller wants to end communication or stop waiting for the buyer, for any of the following reasons: <ul style="list-style-type: none"> Buyer is not a registered user. In this case, the seller can close the dispute immediately. The buyer has not responded within four full days of the dispute being opened (BuyerHasNotPaid), or within seven full days (MutualAgreementOrNoBuyerResponse). The buyer has responded at least once, but the buyer and seller cannot reach an agreement. 	<ul style="list-style-type: none"> The item is eligible for a relist credit. If the buyer is no longer a registered user at the time of the filing, the buyer receives a strike. If the buyer does not respond after four or seven days and the seller chooses to end communication, the buyer receives a strike.
MutualAgreementOrNoBuyerResponse. Either the seller claimed that the buyer agreed to not complete the order line item (transaction) and the buyer subsequently agreed, OR at least seven full days have passed with no response from the buyer to the open dispute.	<ul style="list-style-type: none"> The item is eligible for a relist credit. The dispute is closed. Buyer does not receive a strike. The seller receives a Final Value Fee credit.

Chapter 26

Buyer Disputes

eBay buyers are eligible to open a dispute when an item is not received or when an item arrives but the item is damaged or does not match the description in the item listing. The process for creating an Item Not Received or (Item) Significantly Not As Described dispute will vary, dependent on the eBay site and payment method. Communication between a buyer and seller to resolve an issue is always encouraged by eBay, and is often the fastest and simplest way to resolve a problematic transaction. For tips on working to resolve an issue with a seller, see "[Resolving Buying Issues](#)" on page 442.

In this section:

- ["Buyer Disputes on US, UK and DE Sites" on page 476](#)
- ["Buyer Disputes on Other International Sites" on page 477](#)
- ["Filing Buyer Disputes on PayPal" on page 477](#)
- ["Responding to and Tracking Buyer Disputes Through Trading API" on page 477](#)
- ["General Flow for Buyer Disputes" on page 478](#)

Buyer Disputes on US, UK and DE Sites

The eBay Buyer Protection program and the Resolution Center are currently implemented on the US, UK, and DE sites. For these eBay sites, the Resolution Center is the primary avenue for buyers to create Item Not Received or Significantly Not As Described disputes. The following conditions must be met in order for a buyer to file a dispute in the Resolution Center:

- Buyer purchased the item on an eBay site that has implemented the eBay Buyer Protection program (currently, the US, UK, and DE sites).
- Buyer purchased the item on eBay using the eBay checkout flow or through an eBay invoice.
- The item is eligible for protection under the eBay Buyer Protection program. Ineligible items include eBay Motors vehicle listings, Real Estate listings, Classified listings, Half.com listings, and eBay Digital Music Center items.
- The item was paid for in one lump payment using any of the payment methods integrated with eBay. These payment methods include PayPal, Bill Me Later, ProPay, Moneybookers, Paymate, or a credit card transaction processed through the seller's Integrated Merchant Credit Card account.
- The dispute is filed within 45 days of payment for the item.

The eBay Buyer Protection policy states that a buyer can file an Item Not Received dispute if the item has not within the estimated delivery date range, or 7 days after the payment date if an estimated delivery date range was not provided in the invoice or **Review Your Purchase** page. For damaged items, or items that do not match the listing's description, a Significantly Not As Described dispute can be opened as soon as the buyer receives the item.

For more information on the eBay Buyer Protection program and policies, and the Resolution Center, see the [eBay Buyer Protection home page](#). For more information on retrieving and managing eBay Buyer Protection cases programmatically, see the [Resolution Case Management API User Guide](#).

Buyer Disputes on Other International Sites

The processes for filing Item Not Received or (Item) Significantly Not As Described disputes on other International sites vary. Buyers should check the site's help documentation for more information on buyer issues and disputes.

Filing Buyer Disputes on PayPal

Buyers who purchase an item on eBay using PayPal as the payment method may file an Item Not Received or (Item) Significantly Not As Described dispute using the PayPal Purchase Protection. The PayPal Purchase Protection program is similar to the eBay Buyer Protection program but can only be used when items are purchased through a PayPal account. Log in to your PayPal account and click on the PayPal Purchase Protection link to get more information about the PayPal Purchase Protection program.

Responding to and Tracking Buyer Disputes Through Trading API

Sellers can use the dispute calls of the Trading API to respond to and track legacy Item Not Received or Significantly Not As Described disputes, as well as disputes created through the PayPal Purchase Protection system.

Note: The Resolution Case Management API is used to retrieve, track, manage, and resolve all eBay Buyer Protection Item Not Received and Significantly Not As Described cases that are opened by buyers in the Resolution Center. The Trading API dispute calls cannot be used to track or manage these two case types.

For Item Not Received or Significantly Not As Described disputes that are not created through the eBay Resolution Center, the Trading API dispute calls can be used by the sellers to retrieve and manage these disputes. These calls are described in [Table 26-1](#). These calls are only available to sellers.

Table 26-1 Trading API Dispute Calls

Call	Description
AddDisputeResponse	Allows a seller to respond to or close an Unpaid Item dispute or cancelled transaction. The actions that a seller making this call can take include the following: <ul style="list-style-type: none">■ Respond to or make a comment on the open dispute■ Offer a full or partial refund to the buyer■ Inform the buyer that the actual item or a replacement item has been shipped■ Close a dispute if a mutual agreement has been made between the buyer and seller to cancel the transaction

Table 26-1 Trading API Dispute Calls (Continued)

Call	Description
GetDispute	<p>Retrieves the details of an Unpaid Item dispute or cancelled transaction. The dispute ID is required in the request. Details of the case include the following information:</p> <ul style="list-style-type: none"> ■ The type of case ■ The status of the case ■ Any messages associated with the case
 GetUserDisputes	<p>Retrieves all disputes (except eBay Buyer Protection cases) the requester is involved in as a buyer or seller. The following request filters are available to the caller:</p> <ul style="list-style-type: none"> ■ DisputeFilterType: caller can restrict results based on dispute type or status, including disputes that are closed, or disputes that are awaiting response from a buyer or seller. ■ Date range filter: the ModTimeFrom and ModTimeTo filters can be used to return disputes created within the specified date range. <p>These two filters can be used in conjunction with one another or separately.</p>

General Flow for Buyer Disputes

This section describes a typical flow for settling buyer disputes, regardless of whether the dispute was created through the eBay Resolution Center, PayPal Purchase Protection, or other system. Typically, the process has five steps:

1 The buyer files a dispute

The process for filing a dispute will vary dependent on the system being used. It is a good idea for the buyer to read the help documentation listed on the corresponding site/system to get a better understanding of the policies governing buyer disputes.

2 eBay contacts the seller

Once a buyer dispute is filed, eBay contacts the seller by email about the dispute. The seller can log in to the eBay site and view the buyer's comments or use the [GetDispute](#) call to get details on the opened dispute.

3 The seller makes the first response

The seller can respond to the dispute on the eBay site or through the Trading API. In the Trading API, the seller responds to the buyer using the [DisputeActivity](#) field in the [AddDisputeResponse](#). The following values can be passed in the [DisputeActivity](#) field to respond to the buyer:

- [SellerShippedItem](#)—seller already shipped the original item (applicable to Item Not Received dispute) or a replacement item (applicable to Significantly Not As Described dispute)
- [SellerOffersRefund](#)—seller is offering a full refund (applicable to Item Not Received dispute) or a partial refund
- [SellerAddInformation](#)—seller makes a comment on an Item Not Received dispute
- [SellerComment](#)—seller makes a comment on an Significantly Not As Described dispute

The seller must respond to a buyer dispute within seven days (10 days for an international order) after the dispute is filed.

4 The buyer and seller communicate

At this point, the buyer and seller communicate. The buyer must post comments through eBay's Resolution Center (or other dispute console), but the seller can use either the site or an API call. For eBay Buyer Protection cases, the seller can use the applicable Resolution Case Management call. See "[Disputes and the Resolution Case Management API](#)" on page 446 for more information on Resolution Case Management calls. For legacy or PayPal disputes, the seller can use the Trading API dispute calls (see [Table 26-1](#) on page 477).

5 The dispute is closed or escalated

The buyer can close the dispute at any time after the seller has responded, or if the item has been received for an Item Not Received dispute. If the buyer and seller came to a mutual agreement, the buyer can close the dispute. No action is taken against the seller's account.

If the seller did not respond within seven days (10 days for an international order), or the response and/or action was not adequate for the buyer, the buyer may escalate the case/file a claim. For eBay Buyer Protection or PayPal Purchase Protection disputes, the dispute is escalated to a case. For sites not covered by these two programs, the escalation procedure can vary by site.

6 eBay makes a decision on the case

In some cases, a buyer can escalate a dispute. For escalated eBay Buyer Protection cases, eBay ultimately makes a decision on the case and determines whether the buyer or seller are at fault. Whatever the outcome, the buyer or seller has 45 days to file an appeal to the case decision. See "[Appealing a Case](#)" on page 453.

Part V

Item Specifics and Catalog Products

Working with Custom Item Specifics

481

Chapter 27

Working with Custom Item Specifics

This chapter describes custom Item Specifics and how to work with them in the eBay API. It covers these topics:

- “About Item Specifics” on page 481
- “Retrieving a Listing’s Item Specifics” on page 483
- “Searching Against Custom Item Specifics” on page 484
- “Pre-filling Item Specifics with Product Details” on page 484
- “Checking Category Support for Custom Item Specifics” on page 487
- “Listing with Custom Item Specifics” on page 488
- “Revising and Relisting with Custom Item Specifics” on page 489
- “Using Advanced Features of Custom Item Specifics” on page 490

About Item Specifics

Item Specifics are typical aspects of items in the same category. They are well-known fields that are common to (or “specific” to) all items in a particular category.

Item Specifics enable you to classify items by presenting descriptive details in a structured way. For example:

- If you are listing in a jewelry category, “Total Carat Weight=3.00 carats and larger” could be an Item Specific for a bracelet.
- If you are listing in a Washers & Dryers category, “Type=Top-Loading” could be an Item Specific for a washer.

That is, information like “Total Carat Weight” would be specific to jewelry listings, and information like “Type=Top-Loading” would be specific to Washer & Dryer listings. (Most sellers probably would not specify “Type=Top=Loading” for a bracelet.)

Each Item Specific consists of a name (like “Total Carat Weight”) and a value (like “3.00 carats and larger”).

Figure 27-1 shows some Item Specifics for a gold bracelet listing, similar to how they might appear on eBay’s View Item page.

Figure 27-1 Item Specifics on eBay's View Item Page

Description	
<i>Item Specifics</i>	
Metal & Material : Gold Plated, Filled	Product: Bracelets
Gender : Women's	Gold Type: 14k
Style : Figaro	Total Carat Weight: 3.00 carats and larger
Chain Length : 7 in.	Clasp: Lobster

Buyers and sellers can use Item Specifics to find and compare particular types of items more easily when searching and browsing eBay.

By using Item Specifics in listings, sellers can also benefit in these ways:

- Increased exposure of their listings in eBay's search and browse pages
- Recommendations for more effective names and values (so the seller doesn't have to guess)
- Faster listing times (because you can pre-fill recommended names and values for the seller)
- More structured data to help buyers grasp the details of the seller's listing more easily

In the API, item specifics are name/value pairs (simple strings). A seller chooses from a list of the most popular field names and values defined by other users, and/or defines their own field names and values. In the API, you use the names and values the seller specified (as-is) when you list or display an item.

Item specifics are only supported in certain categories. This chapter includes information on how you use the API to determine these categories.

Item specifics can be included in a listing via three mechanisms:

- You can pre-fill item specifics based on stock information from an eBay product catalog. For example, if you're listing a camera, you just specify the UPC in [Item.ProductListingDetails](#), and eBay will usually be able to fill in all the product details, including the item specifics, for you. See "["Pre-filling Item Specifics with Product Details"](#) on page 484".
- You can define the item specifics yourself in [Item.ItemSpecifics](#). For example, if you're listing an item that has no UPC, ISBN, EAN, or other industry-standard identifier, you can just spell out the details yourself. See "["Listing with Custom Item Specifics"](#) on page 488".
- You can list a collection of similar items as a *multi-variation listing*. For example, if you're listing T-shirts that are all the same brand and style, but they come in different colors and sizes, you can use [Item.ItemSpecifics](#) to define the common details yourself, such as "Brand" and "Sleeve Length", and then use [Item.Variations](#) to specify the item specifics that vary, such as "Color" and "Size", along with individual quantities of each variation that you have available. See "["Overview of Multi-Variation Listings"](#) on page 265".

Retrieving a Listing's Item Specifics

To see custom Item Specifics (if any) that a seller has included in a listing, use [GetItem](#). The item specifics are returned as name/value pairs that you can display to a user. Specify [IncludeItemSpecifics](#) as `true` in the request to retrieve the [ItemSpecifics](#) node. (If the listing has no Item Specifics, the node is not returned.)

[Example 27-1](#) shows a portion of a [GetItem](#) XML response with an [ItemSpecifics](#) node. In this case, the seller listed a gold chain bracelet with a lobster clasp.

Example 27-1 Item Specifics in a GetItem Response

```
<?xml version="1.0" encoding="utf-8"?>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2007-07-09T22:42:31.015Z</Timestamp>
  <Ack>Success</Ack>
  <Version>521</Version>
  <Build>e521_core_Bundled_4938995_R1</Build>
  <Item>
    <Title>Gold Chain Bracelet with Lobster Clasp</Title>
    <PrimaryCategory>
      <CategoryID>110433</CategoryID>
    </PrimaryCategory>
    <!-- more elements here -->
    <ItemSpecifics>
      <NameValuePairList>
        <Name>Metal & Material</Name>
        <Value>Gold Plated, Filled</Value>
      </NameValuePairList>
      <NameValuePairList>
        <Name>Product</Name>
        <Value>Bracelets</Value>
      </NameValuePairList>
      <NameValuePairList>
        <Name>Gender</Name>
        <Value>Women's</Value>
      </NameValuePairList>
      <NameValuePairList>
        <Name>Gold Type</Name>
        <Value>14k</Value>
      </NameValuePairList>
      <NameValuePairList>
        <Name>Style</Name>
        <Value>Figaro</Value>
      </NameValuePairList>
```

Example 27-1 Item Specifics in a GetItem Response

```
<NameValueList>
  <Name>Total Carat Weight</Name>
  <Value>3.00 carats and larger</Value>
</NameValueList>
<NameValueList>
  <Name>Chain Length</Name>
  <Value>7 in.</Value>
</NameValueList>
<NameValueList>
  <Name>Clasp</Name>
  <Value>Lobster</Value>
</NameValueList>
</ItemSpecifics>
</Item>
</GetItemResponse>
```

eBay returns custom Item Specifics in the order that the seller specified them. We suggest you display them in that order, in case the seller grouped or ordered them in a logical way.

Searching Against Custom Item Specifics

eBay indexes popular custom Item Specifics so that keyword queries match against them in Title & Description searches.

The eBay Web site UI provides buy-side finding navigation support for popular custom Item Specifics in select categories.

You can't directly search against custom item specifics using the API, but you can use *aspect filters* in the Finding API to search against most item specifics.

Samples Showing Aspect Histograms and Aspect Filtering in the Finding API
<http://developer.ebay.com/DevZone/finding/CallRef/findItemsAdvanced.html#Samples>

Pre-filling Item Specifics with Product Details

A *product* (or *catalog product*) is a predefined (stock) item description from a catalog. Product information is available for a wide variety of products, such as cameras, books, DVDs, sporting equipment, tickets to well-known events and venues, and more.

Buyers are more likely to buy if they can see key product information immediately. Including details from our product catalog in your listings creates a shopping experience that speeds the buyer's path from search to purchase.

eBay Online Help: Product Details
<http://pages.ebay.com/help/sell/product-details.html>

To buyers, product details are shown in eBay's View Item page, and in similar pages in eBay Mobile apps, as shown in [Figure 27-2](#) on page 485.

Figure 27-2 Product Details in eBay's View Item Page

Seller assumes all responsibility for this listing.

Item specifics		
Condition:	New: A brand-new, unused, unopened, undamaged item in its original packaging (where packaging is ... Read more)	Brand:
Megapixels:	20.2 MP	Model:
Optical Zoom:	3.6x	MPN:
Screen Size:	3"	Type:
Color:	Black	UPC:
Detailed item info		
Product Information		
The Sony DSC-RX100 camera is equipped with the world's first 1.0 type CMOS sensor, with effective resolution of 20.2 megapixels. The Sony Cyber-shot camera boasts of an optimized BIONZ image processing that boosts shooting speed to 0.13 seconds. Even under difficult lighting conditions and up to ISO6400 sensitivity, you will be astonished by the quality of the images. Simply rotate the control ring to choose the function you wish to perform. Functions like zoom control, aperture, and shutter speed are controlled by the 3-inch LCD incorporates a fabulous 1229k dot resolution that lets you see subtle details and tones without any loss of quality.		
Product Identifiers		
Brand	Sony	
Model	DSC-RX100	
MPN	DSC-RX100/B	
UPC	027242856110	
Key Features		
Camera Type	Point & Shoot	
Optical Zoom	3.6x	
Digital Zoom	14x	
Sensor Resolution	20.2 MP	
Screen Size	3"	
Memory / Storage		
Supported Flash Memory	Memory Stick Duo, Memory Stick PRO DUO, SD, SDHC, SDXC	
Dimensions		
Depth	1.41 in.	
Height	2.29 in	

The API supports two main workflows for listing with product details:

- Specify industry-standard identifiers:

For most sellers, the fastest way to list with product details is to pass in an industry-standard identifier, such as ISBN, UPC, GTIN, EAN, or Brand+MPN in [ProductListingDetails](#) in [AddItem](#) or related calls.

Similarly, you can list event tickets by specifying a few well-known details in [ProductListingDetails](#), such as the event name and other data, supplemented by custom item specifics. See “[Ticket Listings](#)” on page 370 for more information on listing tickets.

With this approach, eBay searches its product catalog for a product that matches the ID you passed in, and (if a match is found) uses that product in the listing.

- Search the product catalog for an eBay Product ID (EPID)

With this approach, you use [FindProducts](#) in the Shopping API to search for the right product, and look at its [ProductReferenceID](#) (the EPID).

Or you find a similar item on eBay that was listed with product details, and use [GetItem](#) to retrieve the item's [ProductReferenceID](#).

In either case, once you have the EPID, you pass it in [ProductListingDetails](#) in [AddItem](#) or related calls.

If you use a standard identifier like UPC, eBay will attempt to find a matching product for you, to use in your listing. If a single matching product is found, the item will be listed with that product's stock details, including the product's stock photo and other stock information. If you also set [PictureDetails.GalleryType](#), the stock photo will be used as the Gallery image.

If more than one match is found, eBay returns an error indicating that too many products were found, along with a list of possible product IDs for the products. In this case, either choose one of those IDs, or use [FindProducts](#) in the Shopping API to search for the right product, and use its EPID instead (an EPID is always unique).

It is also possible to list with a value called [ProductID](#) in the API. This is a versioned form of the EPID, and it is less commonly used. However, it may be returned as an option when you use a standard identifier like ISBN, and multiple matching products are found. If you pass in an old product ID that eBay still recognizes and only one match is found, eBay will list the item with the latest version of the product and the latest product ID, and the call will return a warning indicating that the data has changed.

You do not need to specify [PrimaryCategory](#) with [ProductListingDetails](#), because eBay can determine the category based on the product ID. However, if you specify the category, then [PrimaryCategory](#) takes precedence.

ProductID Error Handling

In rare cases, if you pass in an old product ID in [ProductID](#) (or possibly an old EPID), and eBay has mapped the value to a new product ID in the eBay system, you may receive an error or warning that

the product ID was updated. If you encounter this issue, you may be able to control how this error is processed by using the [ErrorHandling](#) field in your request.

Note: eBay makes an effort to minimize such changes to product IDs.

Please also note that if you specify a product ID that eBay does not recognize (including a very old product ID that is not mapped to a current product ID in the system), this is treated as a “product not found” error (12024) instead. eBay’s response to this error condition is not affected by the error handling preference you specify.

Error Messages

Either warning 5119 or error 5123 will be returned, depending on the preference you specify. These are user-facing messages.

Error Handling Preference Effects

The table below shows how eBay handles this scenario based on the error preference you pass in the listing request.

Table 27-1 Catalog Product ID Updated

	BestEffort	AllOrNone	FailOnError	Legacy
Severity	Warning (5119)	Warning (5119)	Error (5123)	Warning (5123)
Listing effect	List the item	List the item	Reject the listing.	List the item.

Common Causes

- A portion of the product data (such as the Additional Information section) changes on eBay. Such a change triggers a new version of the product, which in turn causes a portion of the ID string to change.
- The seller searched for product details previously and is now using the same product ID for another listing, but the ID has changed on eBay since the original product search was performed.
- The seller is relisting an existing item and the product ID has changed on eBay since the item was originally listed.
- The application uses [GetItem](#) to download an existing item with product details and attempts to use the response data in a new listing, but the product ID has changed on eBay since the downloaded item was originally listed.
- Within the application, the seller schedules the item to be listed at a future date, but the product ID changes on eBay by the time the application submits the listing.

Checking Category Support for Custom Item Specifics

You can only list an item with custom Item Specifics if the category you’re listing in supports this functionality.

To determine which categories currently support custom Item Specifics, use [GetCategoryFeatures](#). For the categories (or parent categories) that support custom Item Specifics, it returns [ItemSpecificsEnabled](#) as Enabled.

A few categories may require you to use custom Item Specifics, and a few categories may require you to choose from a list of pre-approved names and/or values. See “[Using Advanced Features of Custom Item Specifics](#)” on page 490 for more information.

Listing with Custom Item Specifics

This section describes how to fill in custom Item Specifics in [AddItem](#) and related calls. This section assumes the seller is listing in one category, and they are filling in their own names and values from scratch (or from their own inventory system or catalog).

Here is the basic approach that you use include custom Item Specifics in a listing:

- 1 When the seller chooses a leaf category to list in, check your local data store to make sure the category supports custom Item Specifics.
If you don't already do so, you should use [GetCategoryFeatures](#) (in combination with [GetCategories](#)) to retrieve and store information about which leaf categories support features like custom Item Specifics. See “[Checking Category Support for Custom Item Specifics](#)” on page 487.
- 2 Provide the seller with fields for entering their own custom Item Specifics.
- 3 Pass the seller's Item Specific names and values in the listing request, along with all the other fields that you would normally include in the request.

When you use [AddItem](#), the seller can specify one value per Item Specific name in listing requests, unless [GetCategorySpecifics](#) indicates that multiple values are supported.

For example, let's assume the seller is listing a gold chain bracelet with a lobster clasp in category 110433. For the custom Item Specifics, the seller is using the following names and values from their own inventory system or catalog:

Table 27-2 Seller's Names and Values

Item Specific Name	Item Specific Value
Gender	Women's
Metal	14k Gold
Chain Style	Figaro
Chain Length	7 in.
Clasp	Lobster

Example 27-2 shows the relevant portion of an `AddItem` request for this listing. As with any listing, it is also a good idea to test the request by using `VerifyAddItem` first.

Example 27-2 Seller's Custom Item Specifics in an AddItem Request

```
<?xml version="1.0" encoding="utf-8"?>
<AddItemRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <ErrorLanguage>en_US</ErrorLanguage>
    <Item>
        <!-- more elements here -->
        <ItemSpecifics>
            <NameValuePairList>
                <Name>Gender</Name>
                <Value>Women's</Value>
            </NameValuePairList>
            <NameValuePairList>
                <Name>Metal</Name>
                <Value>14k Gold</Value>
            </NameValuePairList>
            <NameValuePairList>
                <Name>Chain Style</Name>
                <Value>Figaro</Value>
            </NameValuePairList>
            <NameValuePairList>
                <Name>Chain Length</Name>
                <Value>7 in.</Value>
            </NameValuePairList>
            <NameValuePairList>
                <Name>Clasp</Name>
                <Value>Lobster</Value>
            </NameValuePairList>
        </ItemSpecifics>
        <!-- more elements here -->
        <Title>Gold Bracelet with Lobster Clasp</Title>
        <PrimaryCategory>
            <CategoryID>110433</CategoryID>
        </PrimaryCategory>
    </Item>
    <RequesterCredentials>
        <eBayAuthToken>YOURTOKENHERE</eBayAuthToken>
    </RequesterCredentials>
</AddItemRequest>
```

This is an easy way to include custom Item Specifics in a listing so that the names and values the seller chooses will appear in the listing's View Item page.

To achieve more consistency with other listings on eBay, you can download popular names and values from eBay. See “[Listing with Recommended Item Specific Names and Values](#)” on page 490.

Revising and Relisting with Custom Item Specifics

If the seller changes the listing category when they revise or relist an item, your application needs to specify the `ItemSpecifics` node again. As eBay's recommendations are different for each leaf category, we suggest that you give the seller the option to review and choose from recommended Item Specifics for the new category.

To delete custom Item Specifics when you revise or relist an item, specify “Item.ItemSpecifics” (without quotes) in [DeletedField](#) and do not pass in the [ItemSpecifics](#) node.

As usual with [Reviseltem](#), there are limitations on changing custom Item Specifics when the item is in a restricted (bids/12 hours) state. See the [Reviseltem](#) documentation for the current rules.

Using Advanced Features of Custom Item Specifics

These sections describe some of the more advanced features that are also available to applications that support listing with custom Item Specifics. All of these features are optional.

- [“Listing in Two Categories with Custom Item Specifics” on page 490](#)
- [“Listing with Recommended Item Specific Names and Values” on page 490](#)
- [“Checking Valid String Lengths for Custom Item Specifics” on page 495](#)
- [“Using Standard Units of Measure for Custom Item Specifics” on page 496](#)
- [“Retrieving Recommendations for All Categories on a Site” on page 496](#)

Listing in Two Categories with Custom Item Specifics

This section is only applicable if your application supports listing in two categories.

When you list in two categories, you can use custom Item Specifics as long as at least one of the categories supports this feature. That is, as long as either the primary or secondary category has custom Item Specifics enabled, you can list with custom Item Specifics.

If both categories support custom Item Specifics, you still only specify one [ItemSpecifics](#) node in your listing request. This is because custom Item Specifics are intended to describe the item itself, regardless of the category. eBay will store the names and values in the order you specify them.

Note: If you are also using eBay’s recommended Item Specifics (see [“Listing with Recommended Item Specific Names and Values” on page 490](#)), you can use recommendations from both categories in the same [ItemSpecifics](#) node.

Listing with Recommended Item Specific Names and Values

In most categories, you can list with custom Item Specifics without using eBay’s recommendations. However, there are some cases when an Item Specific is required in a particular category, or you must use a value that eBay has defined (not your own value).

eBay can recommend popular or required names and values for the seller to use for a given item. When you fill in custom Item Specifics, it's a good idea to use well-known or standard names and values, for these reasons:

- Well-known terms can help buyers (and comparison shopping tools) understand the details of your item more easily.
- Standard terms help achieve consistency across items in the same category.
- Recommended Item Specifics are more likely to be indexed for title and description searches in the future.

eBay determines the popularity of a name or value based on several factors, such as the number of recent listings and/or recently sold listings in the same category that have used the same name or value.

Use [GetCategorySpecifics](#) to retrieve the most popular or required Item Specific names and values for each leaf category you request. For each leaf category you specify, this call enables you to check whether the recommendations have changed since you last checked (so that you don't have to download the same data if it hasn't changed). As you can only request recommendations for up to 100 categories at a time, you may need to make several calls to retrieve all the category recommendations. (Alternatively, you can download recommendations for all categories in bulk. See the [GetCategorySpecifics](#) documentation for details.)

When the seller is listing with custom Item Specifics, the seller may be able to list more quickly if you let them choose from a predefined list of recommended Item Specifics, and also give them the option to enter their own names and/or values.

[Figure 27-3](#) illustrates one way to display recommended Item Specifics. Here are some details to note:

- It shows several drop-down lists for recommended Item Specifics (Brand, Product, and so forth).
- Several of the field values are filled with default values based on eBay's top recommendations. For example, the Gender field shows with the value "Women's", and the drop-down for Gender expands to show additional values that are available, plus an option to enter your own value.
- Other names with lower relevance (like Style) are shown as additional specifics. (If selected, another field is displayed with a name and a list of values.)
- There is also an "Add a custom detail" option, which (if selected) enables the seller to enter new custom Item Specific names with values.

Figure 27-3 Entering Custom Item Specifics

The screenshot shows the 'Item Specifics' section of an eBay listing page. It includes fields for Brand, Product, Metal & Material, Solid Gold Karat Weight, and Gender. The Gender field is expanded, showing options like 'Women's' (selected), 'Men's', 'Women's', and 'Enter your own'. A 'Remove' link is visible next to each option. Below the gender field are buttons for 'Add more Specifics' and various category suggestions.

Item Specifics

Enter specifics about your item here to help buyers find your listing.

Brand

Product

Metal & Material

Solid Gold Karat Weight

Gender

Women's

Men's

Women's

Enter your own

Add more Specifics

+ Gold Type + Gemstone & Stone + Style + Total Carat Weight

+ Add a custom detail

You can either download and cache item specifics for all categories, or you can determine which categories your sellers use most frequently, and then download and cache recommendations for just those categories.

GetCategorySpecifics takes up to 100 leaf categories as input, and it returns recommendations for those categories only. To download recommendations for all categories, you need to make multiple calls.

The **GetCategorySpecifics** call reference includes samples that show how to use this call.

GetCategorySpecifics Samples

<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/GetCategorySpecifics.html#Samples>

When you make this call, parse the names and values and the **Timestamp** returned in the **GetCategorySpecifics** response and store them locally.

The recommended Item Specifics are returned in order of popularity, with the most popular name first. For each name, the most popular value is returned first. Therefore, we suggest that you present the recommendations to the seller in the same order in which eBay returns them.

For example, if you are creating a UI for end users, you could display the most popular value (i.e., the first value returned for each specific) by default in each field, and show the remaining values in each field's drop-down list.

Periodically (say, once a day), call [GetCategorySpecifics](#) for that set of categories again and pass the stored [Timestamp](#) value in [LastUpdateTime](#) to see if any data has changed.

If the data for any categories has changed, call [GetCategorySpecifics](#) (without [LastUpdateTime](#)) to refresh the recommendations for those categories. Store the updated data along with the new [Timestamp](#) so that you can check for changes again later.

Once the seller has chosen the names and values they want to use, pass them into [AddItem](#) by using the [ItemSpecifics](#) fields. The seller can specify one value per Item Specific. (If they specify more than one value, eBay uses the first one.)

If the seller doesn't see a name or value they need in the recommendations, the seller can still add their own in the listing. (If that name or value becomes very popular, it is likely to be recommended over time.)

For example, let's suppose the seller is listing a gold chain bracelet with a lobster clasp in category 110433. For the custom Item Specifics, the seller is using several names and values returned from [GetCategorySpecifics](#), plus the seller is adding their own names and values from scratch (say, from the seller's own inventory system or catalog). [Table 27-3](#) shows these Item Specifics.

Table 27-3 Names and Values from eBay and the Seller

Item Specific Name	Item Specific Value	Origin
Metal & Material	Gold Plated, Filled	eBay recommendation
Product	Bracelets	eBay recommendation
Gender	Women's	eBay recommendation
Gold Type	14k	eBay recommendation
Style	Figaro	eBay recommended name. Seller defined value Figaro.
Total Carat Weight	3.00 carats and larger	eBay recommendation
Chain Length	7 in.	Seller defined name and value
Clasp	Lobster	Seller defined name and value

For Chain Length, the seller chose to specify "7 in." instead of "7 inches" because eBay recommends "in." as a preferred unit of measure identifier. (See "[Using Standard Units of Measure for Custom Item Specifics](#)" on page 496.)

Example 27-3 shows the relevant portion of an `AddItem` request with these custom Item Specifics. The request needs to specify all the names and values to include in the listing. (That is, eBay will not automatically add recommended names or values for you.)

Example 27-3 Custom Item Specifics in an AddItem Request

```
<?xml version="1.0" encoding="utf-8"?>
<AddItemRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <ErrorLanguage>en_US</ErrorLanguage>
    <Item>
        <!-- more elements here -->
        <ItemSpecifics>
            <NameValueList>
                <Name>Metal & Material</Name>
                <Value>Gold Plated, Filled</Value>
            </NameValueList>
            <NameValueList>
                <Name>Product</Name>
                <Value>Bracelets</Value>
            </NameValueList>
            <NameValueList>
                <Name>Gender</Name>
                <Value>Women's</Value>
            </NameValueList>
            <NameValueList>
                <Name>Gold Type</Name>
                <Value>14k</Value>
            </NameValueList>
            <NameValueList>
                <Name>Style</Name>
                <Value>Figaro</Value>
            </NameValueList>
            <NameValueList>
                <Name>Total Carat Weight</Name>
                <Value>3.00 carats and larger</Value>
            </NameValueList>
            <NameValueList>
                <Name>Chain Length</Name>
                <Value>7 in.</Value>
            </NameValueList>
            <NameValueList>
                <Name>Clasp</Name>
                <Value>Lobster</Value>
            </NameValueList>
        </ItemSpecifics>
        <!-- more elements here -->
        <Title>Gold Bracelet with Lobster Clasp</Title>
        <PrimaryCategory>
            <CategoryID>110433</CategoryID>
        </PrimaryCategory>
    </Item>
    <RequesterCredentials>
        <eBayAuthToken>YOURTOKENHERE</eBayAuthToken>
    </RequesterCredentials>
</AddItemRequest>
```

Listing with Required Item Specifics

Some categories may require item specifics, or they may require you to use eBay's predefined names or values, rather than using your own. For example, clothing categories typically require you to specify Brand and Color as item specifics. (Color can optionally be specified in variation specifics instead. See "[Overview of Multi-Variation Listings](#)" on page 265.)

To determine which item specifics are required, call [GetCategorySpecifics](#) to retrieve the item specifics recommendations for the category you're listing in. In the response, look for these fields for each recommendation:

- This item specific name is required for the category:
`Recommendations.NameRecommendation.ValidationRules.MinValues`
- You can only use one of these values that eBay has defined (as returned in the response):
`Recommendations.NameRecommendation.ValidationRules.SelectionMode=SelectionOnly`

See "[Listing with Custom Item Specifics](#)" on page 488 for more information about retrieving recommended item specifics in general.

Note that in some categories, item specifics may not be required, but they may be necessary if you want to take advantage of features like parts compatibility. For example, if you are listing tires or rims for vehicles, and you want to enable vehicle owners to easily determine whether your item will match their vehicles, then you need to use eBay's predefined item specifics to specify details like Load Index or Rim Diameter. See "[Listing Items with Parts Compatibility](#)" on page 282 for details about parts compatibility.

Checking Valid String Lengths for Custom Item Specifics

Advanced Topic: This section describes how to check for string length restrictions on custom Item Specifics. This is a more advanced approach to using custom Item Specifics. This feature is optional. You can list with custom Item Specifics without using this information.

The US and international eBay sites may limit the number of Item Specifics in a listing. Use [GeteBayDetails](#) to determine the current maximum number of custom Item Specifics allowed per item, as well as how many characters you can use in a name or value. (The `MaxValuesPerName` field does not apply to listing requests. You can only specify one value per name in listing requests at the time of this writing.)

Example 27-4 shows a portion of the [GetBayDetails](#) response with this information.

Example 27-4 Details for Custom Item Specifics in GetBayDetails

```
<?xml version="1.0" encoding="utf-8"?>
<GetBayDetailsResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2007-07-09T22:30:37.437Z</Timestamp>
  <Ack>Success</Ack>
  <Version>521</Version>
  <Build>e521_core_Bundled_4938995_R1</Build>
  <!-- more elements here -->
  <ItemSpecificDetails>
    <MaxItemSpecificsPerItem>15</MaxItemSpecificsPerItem>
    <MaxValuesPerName>10</MaxValuesPerName>
    <MaxCharactersPerValue>50</MaxCharactersPerValue>
    <MaxCharactersPerName>30</MaxCharactersPerName>
  </ItemSpecificDetails>
  <!-- more elements here -->
</GetBayDetailsResponse>
```

Using Standard Units of Measure for Custom Item Specifics

Advanced Topic: This section describes how to suggest more consistent units of measure to the seller. This feature is optional. You can list with custom Item Specifics without using this information.

For sellers who want to use custom Item Specifics to specify measurements (such as the length of an object), [GetBayDetails](#) provides a set of preferred units to use in value strings. For example, instead of specifying *inches* or " (the symbol), we suggest you use *in.* as a length unit. See [UnitOfMeasurementDetails](#) in [GetBayDetails](#).

Retrieving Recommendations for All Categories on a Site

Advanced Topic: This functionality is mainly useful for applications that support a wide variety of sellers, or sellers who list in a very wide variety of categories.

Normally, [GetCategorySpecifics](#) enables you to retrieve recommendations for 100 leaf categories at a time. This means you need to make multiple [GetCategorySpecifics](#) calls to download data for more categories.

For applications that support a very wide variety of categories, it may be more practical to use the [downloadFile](#) call in the File Transfer API to download a single [GetCategorySpecifics](#) response that contains custom Item Specific recommendations for all categories on a given site. In this case, the [GetCategorySpecifics](#) response is returned as a .zip file attachment. It uses the [Recommendations](#) node

See [GetCategorySpecifics](#) in the *eBay Trading API Call Reference* for details.



Part VI

Searching for and Retrieving Items

Searching for Items	498
Retrieving Items	502

Chapter 28

Searching for Items

On eBay sites, users search for items by filling in fields that are used by eBay for a search query. Items matching the query are displayed in a list, referred to as a *search page*. Having conducted a search, a user can also save the search on the My eBay page.

Important: All search activities that formerly used `GetSearchResults` and `GetCategoryListings` are now performed using Finding API calls. `GetSearchResults` and `GetCategoryListings` are deprecated and are no longer available.

If you have applications that use these calls, please migrate your search functions to the [Finding API](#). For new applications, please start with the Finding API.

This chapter discusses the following search topics:

- “[Searching for Items by Using a Query](#)” on page 498
- “[Searching for Reviews & Guides](#)” on page 499
- “[Finding All Item Listings in a Specified Category](#)” on page 501

For special considerations when searching for listings on the US eBay Motors site, see “[Searching for eBay Motors US Listings](#)” on page 571.

Searching for Items by Using a Query

eBay's API-based searches return data that includes basic item information. They also return data that includes the country and postal code of the item, default shipping cost information, whether PayPal is offered, and store-related information.

An application can enable users to search for items based on user-specified keywords and filter criteria. This is done on eBay using the Finding API. For information about searching for items using a query, please refer to the [Finding API User's Guide](#).

Queries available using the Finding API include the following features:

- Free-text keyword queries, including wildcards. See [Searching by Keywords](#).
- Product ID-based queries, including ID types such as ISBN or UPC. See [Searching by Product](#).
- Product queries without a product ID (Shopping API). See [the FindProducts Call Reference](#).
- Using item filtering. See [Refining a Search with Item Filters](#).
- Using aspect and domain filtering. See [Refining a Search with Aspect and Domain Filters](#).
- Managing the amount and ordering of search results. See [Controlling the Way Search Results are Returned](#).

Searching for Reviews & Guides

Buying Guides are available in the Production Environment only. They are not available in the Sandbox.

When user searches for items in certain categories, the results may also include URLs for relevant buying guides.

Buying guides contain content about particular product areas, categories, or subjects to help buyers decide which type of item to purchase based on their particular interests. These guides are intended for buyers who do not already have a specific product in mind. For example, a digital camera buying guide could help a buyer determine what kind of digital camera is right for them.

For a list of buying guides that are currently available on the US site, see the Buying Guide hub:

Buying Guide Hub (US site)
<http://reviews.ebay.com>

Search queries may return URLs for multiple buying guides. Buying guides may be returned for a matching item's category as well as its parent and child categories. Information about each buying guide is returned in the same order that it would be displayed on the Web site. The guide for the current category (if any) is returned first. Then guides for its child categories are returned. Finally, guides for its parent categories are returned. When many buying guides are found, only URLs for the first few buying guides are returned. Use the URLs along with the buying guide names (also returned) to create links in your application to the buying guides.

You can retrieve buying guide details by using the Shopping API **FindProducts** call to find a product, then retrieve the product's reviews and guides (see “[Searching for Products, Reviews, and Guides](#)” on page 499).

Buying guide details are not necessarily returned for all searches, as some categories and product finders are not associated with buying guides.

To test your application's ability to correctly present functional buying guide links, you can begin by testing against the US site. For example, you can search in these categories on the US site (use **GetCategories** to retrieve the latest category IDs):

- Digital Cameras (category ID 29997)
- Jewelry & Watches (category ID 281)
- Baby (category ID 2984)

Searching for Products, Reviews, and Guides

In addition to searching for listings, you can also search eBay's product catalog for products.

Product searches can be useful to applications that support shopping comparison, product reviews, or basic supply and demand data.

To search for products, use the [Shopping API FindProducts call](#). One common case is to start by specifying a keyword query. This retrieves a list of products with no listings, reviews, or guides. This is analogous (although not identical) to using this page on the eBay Web site:

Children's Books Product Search
http://product-search.ebay.com/Childrens-Books_W0QOpovcsZ1389

With [FindProducts](#), you can retrieve up to 2000 products that match a query. Each product includes basic details, such as the product title and Item Specifics. (Item Specifics are standardized details that are specific to the category. For example, a book's Item Specifics are fields like author, format, publication year, and ISBN. A ticket's Item Specifics are fields like Venue, Event Type, and Number of Tickets.)

Note: Some product content is provided to eBay by third-party providers. This means the API cannot return certain information, such as the detailed description text that is owned by the provider. Instead, we provide a link to the product detail page hosted on eBay.

For each product, you can retrieve additional information that would be useful to buyers, including top reviews by eBay members, buying guides (shopping advice), and up to 200 matching items.

The Shopping API [FindProducts](#) call also supports affiliate tracking for members of the eBay Affiliates Program. See “[Affiliate Tracking Concepts](#)” on page 155. eBay Affiliates earn commissions for driving traffic to eBay.

Presenting Links to Buying Guides

Buying Guides are available in the Production Environment only. They are not available in the Sandbox.

When user browses for items in certain categories, URLs for multiple buying guides can be returned (see [BuyingGuideDetails](#)).

Buying guides contain content about particular product areas, categories, or subjects to help buyers decide which type of item to purchase based on their particular interests. These guides are intended for buyers who do not already have a specific product in mind. For example, a digital camera buying guide could help a buyer determine what kind of digital camera is right for them.

Buying guide details can be returned based on the category specified in the request as well as its parent categories and subcategories. In this case, the buying guide details are returned in the category's browse order. For example, the guide for the current category (if any) might be returned first, followed by guides for its child categories, and then guides for its parent categories. When many buying guides are found, only URLs for the first few buying guides are returned. Use the URLs along with the buying guide names (also returned) to create links in your application to the buying guides.

Buying guide details are not necessarily returned for all categories, as some categories are not associated with buying guides.

To test your application's ability to correctly present functional buying guide links, you can begin by testing against the US site. For example, you can retrieve listings in these categories on the US site (use [GetCategories](#) to retrieve the latest category IDs):

- Digital Cameras (category ID 29997)
- Jewelry & Watches (category ID 281)
- Baby (category ID 2984)

Finding All Item Listings in a Specified Category

An application can retrieve the item listings for a particular category on an eBay site. It can also retrieve a list of the categories for a particular eBay site, from which the end-user may select a category. In addition, category search can apply filters to refine the results, sort the results by relevance or another criterion, and specify the data quantity and format to return.

Searching by category is done on eBay using the Finding API. For information about searching for items by category, please refer to the [Finding API User's Guide](#).

Chapter 29

Retrieving Items

This chapter discusses the means for retrieving the data for item listings.

An application can retrieve data from the eBay site, to be displayed in the application's particular user interface or otherwise processed by the application. Items can be retrieved in whole groups (such as a list of the items a particular seller has listed). This replicates similar functionality on the eBay site. Items can also be retrieved one at a time, which is analogous to the View Item page on the eBay site.

One reason an application returns the data for an item or group of items is to display the item data in a visual format. The application's end user could then see the data and use the information as the basis for taking some action, such as a seller revising the item or a buyer making a bid on the item.

Another reason for retrieving item data is the programmatic processing of item data outside the context of a user interface or human intervention. For example, an application could retrieve a list of the items for a particular seller for the purpose of assessing bidding activity, only alerting the seller should a predetermined condition exist (like the end of the auction is near and there is a current high bidder).

Applications retrieve whole lists of items for a particular seller using [GetSellerList](#). Retrieving the data for just a single item is done with [GetItem](#). Most often it is more economical to use [GetSellerList](#) to retrieve a list of items and operate on them than it is to use [GetItem](#) multiple times. Doing this saves on the number of API calls made by an application (which are limited on an hourly, daily, and monthly basis). It also reduces total online time spent communicating and interacting with the eBay site.

["Browsing a Seller's Items" on page 502](#)

["Retrieving the Data for One Item" on page 504](#)

["Displaying Discount Pricing Information to Buyers" on page 505](#)

For information on [GetSellerList](#), including executing the call and business logic, see the chapter ["GetSellerList" on page 859](#). For information on [GetItem](#), see ["GetItem" on page 833](#). Refer to the [eBay Trading API Call Reference](#) for descriptions of the call-specific input and output fields.

Browsing a Seller's Items

To retrieve a list of the items listed by a particular seller, use [GetSellerList](#).

[GetSellerList Call Reference](#)

<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/GetSellerList.html>

In all calls, at least one date-range filter must be specified.

You can specify a **DetailLevel** to control the response data. If you set a **DetailLevel**, you must set pagination values. Specify the eBay user ID for the seller whose item's will be returned. This input is optional. If it is not specified, the user making the request is considered to be the seller whose items are returned.

The number of items that could be returned for a given seller is potentially large. To reduce the number of items returned, use date-based filter inputs. These filter the data based on when the items started, when the items ended, or when the items were last modified. Applications use this filtering to limit the items to just those of interest in the context of a given operation. For example, if the context calls for retrieving just the items that end on a particular day, the application should not filter for all the items ending within a given month.

Date-based filters work in pairs, delineating a date range described by the earliest (oldest) date of interest and a the latest (most recent) date of interest. For example, if you are retrieving the items that end on a particular day, the earliest date in the range might be one second after midnight on the morning of interest, and no items that ended prior to that date would be of interest.

The latest date in the range might be midnight that night, and no items that ended after that date would be of interest. When one filter input is used, the other matching filter element must also be used (the application must always specify *both* the oldest date and the most recent date for a filtering date range). For example, if **EndTimeFrom** is specified (because the filtering date range is based on the end date) then **EndTimeTo** must also be specified. At least one date-range filter must be specified. Multiple date-range filters may be used per call. For instance, a single call can return the items ending within a date range (one filter) plus those that started in a specified date range (a second filter).

Note: Date-based filters are individually optional, but at least one must be specified (applicable to all detail levels).

Another way to reduce the number of items returned in any single call is through pagination. Pagination breaks the total data set down into smaller chunks that might be easier for the application to use.

Note: If zero items would be returned, no “Invalid page number” error will be returned.

For example, if there are 100 items for a seller meeting other selection criteria for the call, the application could make one call retrieving all 100 items or ten calls that each retrieve 10 items (ten pages of ten items each).

Where the **GetSellerList** inputs are concerned, pagination is controlled by specifying the number of items to return in each page of data (i.e., for each call) and the page number to return. The first call would return page 1 of the seller's items. Data in the call's result set indicate whether there are additional items to retrieve and how many. The application can use that information to determine whether additional calls need to be made and, based on the page size (the number of items per page) how many additional calls need to be made for all of the items to be retrieved. If additional calls are needed, each subsequent call specifies an incrementally higher page number to retrieve each succeeding page of items. If items per page and page number are not specified for the call, no pagination takes place and the call attempts to retrieve the items as a single page of data.

Note: Pagination is required if a detail level is specified. Only by using the smallest result set (no detail level specified) can the call be made without pagination—regardless of how many items would be returned.

The items retrieved using [GetSellerList](#) can optionally be sorted, based on the listing end dates of the returned items. It returns the items the seller has listed.

There is one [ItemType](#) object for each item listed by the specified seller and meeting the input filtering criteria. Also returned are indicators of how many items are returned and whether there are other items that could be returned (additional items meeting the criteria that require additional calls to retrieve). The result set also contains the page number for the data returned, the number of items being returned per page, and the total number of pages of items meeting the selection criteria.

The application needs to traverse this item array to visit each of the [ItemType](#) objects it contains and inspect its properties.

When a listing includes catalog product details and the seller has chosen to include additional stock summary information in the listing, the [Description](#) field is optional. This means items returned by [GetSellerList](#) (and other item-retrieval calls) are not guaranteed to contain a value in the [Description](#) field. See “[Pre-filling Item Specifics with Product Details](#)” on page 484 for more information about listing with catalog product details.

Retrieving the Data for One Item

To retrieve the data for a single item, use [GetItem](#). Doing this involves three general steps: setting up the execution environment, specifying the data to return (based on the item ID), and making the API call.

Retrieving the data for a single item entails specifying the item ID for the item to return.

[GetItem](#) returns the data for one item. The result set consists of one [ItemType](#) object, which contains the item’s data. See the [GetItem](#) documentation in the *eBay Trading API Call Reference* for details.

GetItem Call Reference

<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/GetItem.html>

Retrieving an Item with Item Specifics and/or Product Data

If a listing includes Item Specifics, the Item Specifics will be returned when you execute [GetItem](#) with the [ReturnAll](#) or [ItemReturnAttributes](#) detail level. Similarly, if the listing includes catalog product details, the product details will be returned with these detail levels. See “[Working with Attribute-Based Features](#)” on page 524 for information about listing with Item Specifics and product details.

When you list in certain categories (or, more specifically, certain characteristics sets), [GetItem](#) may return additional Item Specifics that are computed from other attribute values. This means that the [AttributeSet](#) object in the response may not always be identical to the [AttributeSet](#) that the seller submitted. For example, suppose a category’s Item Specifics include separate attributes for City (e.g., “San Jose”), State (e.g., “CA”), and Zip Code (e.g., “95125”). In some categories, the [GetItem](#) response might contain those attributes plus an additional attribute that shows all three values together in one field (e.g., “San Jose, CA 95125”) for display purposes.

When a listing includes product details and the seller has chosen to include additional stock summary information in the listing, the **Description** field is optional. This means items returned by **GetItem** (and other item-retrieval calls) are not guaranteed to contain a value in the **Description** field.

The Item Specifics data in the **GetItem** response includes attribute and value IDs, and literal values for the specified IDs. If your application has saved the attribute meta-data returned from **GetAttributesCS** that was in effect at the time of the listing start date, you can use the meta-data to determine the display name for each attribute.

Displaying Discount Pricing Information to Buyers

The Strikethrough Pricing and Minimum Advertised Price display treatments are available to only those sellers who have qualified to participate in the Discount Pricing program. If you are interested in becoming a member of this program, please contact your account manager or your Customer Service representative.

Some eBay fixed-price listings have discount pricing information, which specifies the original retail price of the item along with additional information about the visual treatment for the seller's discount price. The following sections describe the following two visual treatments of price information and their proper use:

- Strikethrough Pricing (STP)
- Minimum Advertised Price (MAP)

Important: For listings with Minimum Advertised Price details and a seller-specified price below the minimum advertised price, you are legally required to follow the rules for displaying the price of the item to potential buyers. You are bound by the terms of the API License Agreement to follow these rules. Refer to the API License Agreement for consequences of non-compliance.

Because of the legal implications of the Minimum Advertised Price treatment, applications that show item prices to buyers must first check items for discount price information (e.g., **Item.DiscountPriceInfo**) and the specific price treatment (e.g., **Item.DiscountPriceInfo.PricingTreatment**) to determine whether a special display treatment is needed for the item price. Discount price information is returned by the Finding API search calls, **GetSingleItem** and **GetMultipleItems** in the Shopping API, and **GetItem** in the Trading API.

Rules and guidelines for the display of discount pricing information apply to both active and completed items.

Note: There are no specific pricing display rules associated with promotional sales as specified by Markdown Manager or the related API calls (e.g., **SetPromotionalSale**). If either Strikethrough Pricing or Minimum Advertised Price is specified for a listing, it supersedes any promotional sale treatment.

Display Guidelines for Strikethrough Pricing

Strikethrough Pricing is available on the eBay US, eBay UK, and eBay Germany (DE) sites. Each of these sites have different display treatments for items that are listed with **DiscountPriceInfo** elements.

Strike Through Pricing is a visual treatment, consisting of an “original” price in a strikethrough font, as well as the current discounted item price in a non-strikethrough font. The original price is the price (excluding shipping and handling fees) at which the same item, or one that is virtually identical to it, is being offered for sale. On the UK and Germany (DE) sites, the seller can also specify that they have offered a similar item for sale on or off of eBay.

Note: The US, UK, and DE sites all use different notations and phrasing for their strike-through pricing display treatments. For example, The US site makes use of the phrase “Compare at”, while the UK site uses “RRP”, and the DE site uses “UVP”.

Determining Applicability

The display rules for Strikethrough Pricing apply when an item is listed with discount pricing information that results in **PricingTreatment** to return **STP**. To qualify for an STP display treatment, the item **StartPrice** price must be below the **OriginalRetailPrice**, as shown in the following snippet from an XML response from the Trading API:

Example 29-1 API response sample for Strikethrough Pricing

```
...
<StartPrice currencyID="USD">149.99</CurrentPrice>
...
<DiscountPriceInfo>
    <OriginalRetailPrice currencyID="USD">159.99</OriginalRetailPrice>
    <PricingTreatment>STP</PricingTreatment>
    <SoldOneBay>true</SoldOneBay>
</DiscountPriceInfo>
...
```

The response XML for discount pricing information may vary slightly from API to API.

Treatment Rules

You are not required to use the Strikethrough Pricing treatment for listings that have strikethrough pricing information. You can simply display the item price without any special treatment.

If you choose to provide a Strikethrough Pricing treatment, the application should display the “original” price (**Item.DiscountPriceInfo.OriginalRetailPrice**) in a strikethrough price along with the BIN item price (**Item.StartPrice**) in a non-strikethrough font, and text to indicate there is a discount (for example, “Compare at”). The eBay UK and eBay Germany (DE) sites have slightly different discount display treatments, as described in the sections below.

If the item is listed on the UK or German (DE) sites, item can also include a **SoldOneBay** or **SoldOffeBay** field that, if used, indicates that the same seller offered a similar item for sale within the last 60 days at a price indicated by the **OriginalRetailPrice** value.

The following image provides an example of the Strikethrough Pricing visual treatment on the US site. eBay includes a footnote, indicated by the asterisk, to explain that the “Compare at” price comes from another online retailer, not eBay.

Apple iPod nano 3rd Generation Silver (4 GB)

Buy new : \$149.99 | Compare at*: \$159.99

67 new from: \$87.99 | 53 refurbished from: \$75.99 | 122 used from: \$55.99

Next auction ends: \$33.80 (26m)

★★★★★ (533 user reviews)

254 items available from all eBay sellers

When applying a Strikethrough Pricing treatment for an eBay listing, your application should display or include a link that displays explanatory text on pages upon which item price is hidden. The explanatory text in the US should read as follows:

A “Compare at” price is the price (excluding shipping and handling fees) this seller has provided at which the same item, or one that is virtually identical to it, is being offered for sale or has been offered for sale within the last 90 days.

Modify this text, as needed, for discounted items listed in the UK or Germany (DE).

Strikethrough Display Treatments on the UK Site

The eBay UK site employs the following strikethrough treatment styles:

- If an item qualifies for Strikethrough pricing, the item display shows **RRP** followed by a strikethrough price. Below this, **You save** displays with the amount saved in British pounds:

Item condition: --

Time left: 4d 23h (27 Jul, 2011 19:38:38 BST)

RRP: £12.00

You save: £2.00 (16%)

£10.00 **Buy it now**

Add to Watch list ▾



- If the item has been sold by the same seller *on* eBay during the past 30 days, the item display shows **Was** followed by a strikethrough price. Below this, **You save** displays with the amount saved in British pounds:

Item condition: --

Time left: 4d 23h (27 Jul, 2011 19:35:16 BST)

Was: £12.00

You save: £2.00 (16%)

£10.00 **Buy it now**

Add to Watch list ▾



- If the item has previously been sold by the same seller *off* of eBay during the past 30 days (either on another web site or offline store), the item display shows **Was*** followed a strikethrough price. Below this, **You save** displays with the amount saved in British pounds:

Item condition: --

Time left: 4d 23h (27 Jul, 2011 19:37:28 BST)

Was*: £12.00 ?

You save: £2.00 (16%)

£10.00 **Buy it now**

Add to Watch list ▾



Strikethrough Display Treatments on the Germany (DE) Site

The eBay Germany (DE) site employs the following strikethrough treatment styles:

- If an item qualifies for Strikethrough pricing, the item display shows **UVP** followed by a strikethrough price. Below this, **Sie sparen** displays with the amount saved in euros:

The screenshot shows an eBay Germany item listing for a new item. The item is listed at EUR 10,00. The original price (UVP) was EUR 12,00, which is displayed in red with a strikethrough line through it. Below the price, the text "Sie sparen" is followed by "EUR 2,00 (16%)". A blue "Sofort-Kaufen" button is visible.

- If the item has been sold by the same seller *on* eBay during the past 60 days, the item display shows **Ursprünglich** followed by a strikethrough price. Below this, **Sie sparen** displays with the amount saved in euros:

The screenshot shows an eBay Germany item listing for a new item. The item is listed at EUR 10,00. The original price (Ursprünglich) was EUR 12,00, which is displayed in red with a strikethrough line through it. Below the price, the text "Sie sparen" is followed by "EUR 2,00 (16%)". A blue "Sofort-Kaufen" button is visible.

- If the item has previously been sold by the same seller *off* of eBay during the past 30 days (either on another web site or offline store), the item display shows **Was*** followed a strikethrough price. Below this, **Sie sparen** displays with the amount saved in euros:

The screenshot shows an eBay Germany item listing for a new item. The item is listed at EUR 10,00. The original price (Was*) was EUR 12,00, which is displayed in red with a strikethrough line through it. Below the price, the text "Sie sparen" is followed by "EUR 2,00 (16%)". A blue "Sofort-Kaufen" button is visible.

Display Rules for Minimum Advertised Price

Minimum Advertised Price is supported on the eBay US site only.

While sellers have the freedom to set item prices as they see fit, some manufacturers restrict how prices on certain items can be displayed or communicated to others. When a listing includes Minimum Advertised Price (MAP) discount pricing information, there are legal requirements for the display of the item price to buyers when it is below the minimum advertised price set by the manufacturer. If a seller prices an item below the minimum advertised price, applications cannot display the price on any page until the buyer takes further action (such as clicking a button or link).

If the item price is greater than or equal to the minimum advertised price, no special treatment is needed to display the price.

Determining Applicability

The display rules for Minimum Advertised Price apply when an item is listed with discount pricing information that results in [PricingTreatment](#) to return [MAP](#). To qualify for a MAP display treatment, the item [StartPrice](#) price must be below the [MinimumAdvertisedPrice](#), as shown in the following snippet from an XML response from the Trading API:

Example 29-2 API response sample for Minimum Advertised Price

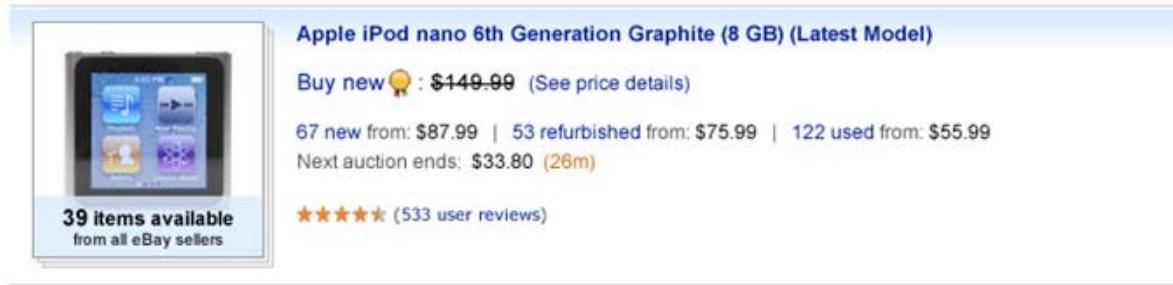
```
...
<StartPrice currencyID="USD">149.99</CurrentPrice>
...
<DiscountPriceInfo>
    <OriginalRetailPrice currencyID="USD">159.99</OriginalRetailPrice>
    <MinimumAdvertisedPrice currencyID="USD">154.99</MinimumAdvertisedPrice>
    <MinimumAdvertisedPriceExposure>DuringCheckout</MinimumAdvertisedPriceExposure>
    <PricingTreatment>MAP</PricingTreatment>
</DiscountPriceInfo>
...
```

The response XML for discount pricing information may vary slightly from API to API.

Treatment Rules

When the display rules for Minimum Advertised Price apply, the application cannot display the seller's lower price until the buyer takes further action. If the exposure setting ([Item.DiscountPriceInfo.MinimumAdvertisedPriceExposure](#)) is [PreCheckout](#), the buyer can click a link or button to view the item's price in a pop-up window or subsequent page. If the exposure setting returns [DuringCheckout](#), the price is displayed on eBay's Review and Confirm page in the checkout flow, or a comparable page in a third-party application that supports checkout capabilities, such as [PlaceOffer](#). If your application sends buyers to eBay for checkout, your application should not display the item price to buyers when the exposure setting is [DuringCheckout](#).

The following image shows a permissible visual treatment when item price is below the specified minimum advertised price. Users must click the “See Price Details” link to view the item price.



When applying a Minimum Advertised Price treatment for an eBay listing, your application must display or include a link to display the following explanatory text on pages upon which item price is hidden:

Sellers have the right to set their own prices independently, but some manufacturers place restrictions on how those prices may be displayed or communicated to others. Because the seller's price on this item is below the manufacturer's "minimum advertised price" the manufacturer may not allow the seller to show you the seller's lower price until you take further action, such as clicking on a drop-down menu, placing the item in your cart, or proceeding to checkout. The steps required depend on the terms of the manufacturer's minimum advertised price policy. Taking these steps allows eBay to show you the seller's great price consistent with eBay's goal of always bringing consumers like you the lowest possible prices on the widest selection of products.

You should know that by taking these steps, you won't be required to purchase the product. You can always simply remove the item from your cart or not proceed with checkout if you decide not to buy the item.

These display rules for discount pricing apply to both active and completed items.

Additional Discount Pricing Information

For more information about discount pricing, see the following resources:

[DiscountPricelInfo](#) description in the Finding API Call Reference

[DiscountPricelInfo](#) description in the Shopping API Call Reference

[DiscountPricelInfo](#) description in the Trading API Call Reference

Part VII

Retrieving User Data

My eBay

525

Managing User Information

513

Chapter 30

Managing User Information

This chapter discusses the means for working with and managing various types of user information, from the data for an eBay user to a user's preference settings.

Every person who participates in an item auction—as a seller, a buyer, or a bidder—must first be a registered eBay user to do so. Registering as an eBay user confers on that person a unique identifier by which that person is known in the eBay environment. Additionally, various properties store information about the user, such as the date the user registered on eBay and whether that user is also an eBay Store owner. These user properties can be retrieved by another user, on the eBay site or through an application. (For more information about users, see “[Users](#)” on page 145 and “[eBay User Roles](#)” on page 147.)

Users can control many aspects of their eBay experience through preference settings. These settings are made on the eBay site, on a user's My eBay page. A number of important settings can also be made from an application. One example of a user preference is a seller's decision whether to have his or her other items cross-promoted when one of their items is viewed or sold.

All eBay sellers have payment accounts at eBay. These accounts are the vehicle through which a seller pays for items listed and sold on eBay. A user may retrieve account information via the eBay site and account information can also be retrieved by an application. For information on retrieving email addresses, see “[Email Privacy Policy](#)” on page 524.

eBay Stores allows a seller to set up a virtual, online storefront in which that seller conducts business. In setting up a store, the seller defines custom categories, in which that seller may list items that prospective buyers see in and purchase from the storefront. These custom categories are Store details that can be retrieved by an application. To list an item in a Store, the seller must know the ID for the custom category in the Store. An application can retrieve this data and supply it for selection by the seller when the seller is in the process of defining a new item.

- “[Retrieving User Information](#)” on page 513
- “[Managing User Preferences](#)” on page 517
- “[Getting Account Information](#)” on page 521
- “[Getting Information about a User's Store](#)” on page 524
- “[Email and Address Privacy Policy](#)” on page 524

For information on [GetUser](#), including executing the call and business logic, see the chapter “[GetUser](#)” on page 876.

Retrieving User Information

To retrieve the data for a particular eBay user, use [GetUser](#). Doing this involves three general steps: setting up the execution environment, specifying the data to return (based on the target user's eBay user ID), and making the API call.

Indicate the user whose data is to be returned by assigning the eBay user ID for that user to the **UserID** property of the object. Certain private data (such as email addresses) is normally not returned with call responses. See “[Email Privacy Policy](#)” on page 524. To indicate this relationship, pass the item ID of the successfully concluded auction in the **ItemID** property.

Table 30-1

Item Property	Required	Type/Code List	Purpose
UserID	N	string	eBay user ID for the user whose data is being returned. If not specified, then the user data returned is for the requestor.
ItemID	N	string	Item ID for a listing in which both the requestor and the target user participated (as seller or buyer). Needed to return certain information otherwise hidden, like the email address.

[GetUser](#) returns a **User** object that contains the data for the specified user. An application needs to inspect the properties of this object to get individual data values. See the documentation for **UserType** for a complete list of its properties.

Table 30-2

Returned Values	Type/Code List	Meaning
User	UserType	User object containing the data for one user.

Uniquely Identifying a User

While the value returned in the **User.UserID** property uniquely identifies an eBay user, this value cannot be relied on 100% of the time. This is because the user may choose to change his or her eBay user name. When this happens, applications can not rely on the **User.UserID** value to associate the user with such things as item listings and item sales.

To uniquely and consistently identify a user, use the value returned in **User.EIAToken**. The EIAS value for a given user does not change, even if the user's eBay user name changes. An application stores this value the same way it stores the eBay user name. When making calls that return a **User** object (which contains both the **UserID** and **EIAToken**), the application checks the user ID returned against the one it has stored and changes the stored user ID as necessary. This check is done based on the **EIAToken** property. For example, if the application stores the data in a database, it would query the database for the **EIAToken** value to find the right record and compare the eBay user ID values.

Note that the unique user identifier returned in the **EIAToken** property has nothing to do the authentication token that you pass in API calls. It also cannot be used as input in places that call for an eBay user ID (such as the **UserID** input property for [GetUser](#)). It does serve as a way to connect user references that have different eBay user names in returned user data. For example, [GetItem](#) returns information about the item's current high bidder. Assuming two different item listings where the same user is the high bidder and where that user changed eBay user names between bidding on the two listings, then the value returned in the **EIAToken** property would allow the application to associate the two with the same buyer and to identify both bidders as the same user.

Working with Anonymous User Information

To protect bidders from fraud, information about a user is made anonymous during bidding. The following rules apply:

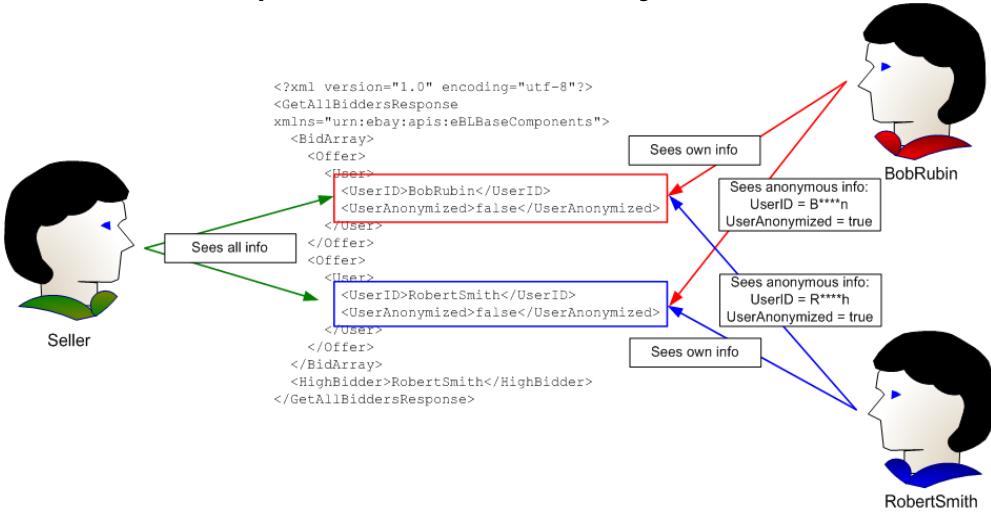
- Bidders can see their own information
- Information is made anonymous in other cases

When the auction has ended, the User ID of the high bidder is no longer anonymous.

The following API calls may return data that has been made anonymous, depending on who is making the call:

- GetAllBidders
- GetBidderList
- GetItem
- GetItemTransactions
- GetMyeBayBuying
- GetSellerEvents
- GetSellerList
- GetSellerTransactions
- GetUser

Figure 30-1 An Example of User ID Made Anonymous for Different Users



During bidding:

- The `UserAnonymized` tag is returned as false to the bidder
- The `UserAnonymized` tag that pertains to a bidder is returned as false to the seller of the item

For other users, this tag is returned as true.

In most cases, a field is made anonymous by removing it from the response. Some fields are made anonymous by replacing the real value with a value that doesn't reveal anything about the user:

- **FeedbackScore:** Replaced by the value "-99".
- **UserID when bidding on items: UserID.** In bidding situations (such as when listing the high bidder in an auction), eBay replaces the `UserID` value with an anonymous bidder name value, such as "a***o".

Note that in this format, the anonymous bidder ID stays the same for a given auction, but is different for different auctions.

- **UserID (GetMyeBayBuying only) when bidding on items: UserID** is replaced with the string "High Bidder".

The Call Reference identifies the fields that are made anonymous for a given call.

Managing User Preferences

On the eBay site, user preferences can be viewed and set on a particular user's My eBay page. In an application, managing user preferences is accomplished with [SetUserPreferences](#) and [GetUserPreferences](#). User preferences that can be get or set include the following:

- Rates options
- Combined purchase preferences
- Flat shipping preferences
- Calculated shipping preferences
- Cross promotion preferences

By using calls to manage a user's preferences, an application can improve the user experience, because the user does not have to be redirected to the eBay site to change certain of his or her preferences. Note that for privacy, legal, or other reasons, some user preferences can only be updated by the user through the eBay site.

Getting User Preferences

Applications can get user preferences with [GetUserPreferences](#). The call takes a set of boolean values as input that define whether each category of user preferences is returned.

Once you have made the call, you can retrieve the user preferences from the response. You can display the current set of preferences to the user, and you can use modify the same preferences object or objects and use them as input to set the preferences by using [SetUserPreferences](#). The next section provides more information on the individual preferences and discusses how to set the user preferences.

Setting User Preferences

Use [SetUserPreferences](#) to set the user preferences. For more information on this call, see “[SetUserPreferences](#)” on page 940.

Examples of the preference containers available in this call are the following:

- [SellerPaymentPreferences](#)
- [CombinedPaymentPreferences](#)
- [CrossPromotionPreferenes](#)

Each preference container has properties. The following tables contain examples of some of the properties:

Table 30-3

Property	Type	Purpose
DisplayPayNowButton	DisplayPayNowButtonType	Specifies whether to display the “Pay Now” button in the seller’s listings, and if so, whether to show it for all payment methods or only for PayPal
PayPalPreferred	boolean	Specifies whether to indicate to buyers that the seller prefers payments via PayPal.
UPSRateOption	UPSRateOptionCodeType	Specifies whether to use the daily rates or the on-demand rates for UPS.

Table 30-4

Property	Type	Purpose
CalculatedShippingPreference	CalculatedShippingPreferencesType	Contains various properties that specify calculated shipping options
CombinedPaymentOption	CombinedPaymentOptionCodeType	Specifies whether the seller accepts combined payment for multiple items, and whether a discount is provided up front or later.
CombinedPaymentPeriod	CombinedPaymentPeriodCodeType	Specifies the period within which payment for multiple items can be combined.
FlatShippingPreferences	FlatShippingPreferencesType	Contains properties that specify flat shipping cost preferences

Table 30-5

Property	Type	Purpose
CalculatedShippingAmountForEntireOrder	AmountType	The calculated shipping amount for the entire order.
CalculatedShippingChargeOption	CalculatedShippingChargeOptionCodeType	Specifies whether to charge the actual shipping and handling cost for the entire order or for each item.
CalculatedShippingRateOption	CalculatedShippingRateOptionCodeType	Specifies whether to calculate the actual rate shipping based on the combined item weight or the individual item weight.
InsuranceOption	InsuranceOptionCodeType	Specifies whether or not to offer or require insurance for calculated shipping.

Table 30-6

Property	Type	Purpose
AmountPerAdditionalItem	AmountType	If the user specifies <code>ChargeAmountForEachAdditionalItem</code> for <code>FlatShippingRateOption</code> , then this value is used to specify how much to charge for each additional item.
DeductionAmountPerAdditionalItem	AmountType	If the user specifies <code>DeductAmountFromEachAdditionalItem</code> for <code>FlatShippingRateOption</code> , then this value is used to specify how much to deduct for each additional item shipped.
FlatRateInsuranceRangeCost	FlatRateInsuranceRangeCostType	For the given range, specifies the cost of insurance. If the <code>InsuranceOption</code> is either <code>Optional</code> or <code>Required</code> , then 6 instances of this field must be specified, one for each cost range.
FlatShippingRateOption	FlatShippingRateOptionCodeType	Specifies how the user wants flat-rate shipping to be calculated.
InsuranceOption	InsuranceOptionCodeType	Specifies whether insurance is offered for flat-rate shipping.

Table 30-7

Property	Type	Purpose
FlatRateInsuranceRange	FlatRateInsuranceRangeCodeType	The price range for the shipment for which the insurance cost is being specified.
InsuranceCost	AmountType	The cost of insurance for the specified price range.

Table 30-8

Property	Type	Purpose
Enabled	boolean	Specifies whether cross-promotions are enabled for the seller's listings.
CrossSellItemFormatSortFilter	ItemFormatSortFilterCodeType	(eBay Store owners only) Specifies what items to show for cross-promotions.
CrossSellGallerySortFilter	GallerySortFilterCodeType	(eBay Store owners only) Specifies Gallery-related options for cross-promotions
CrossSellItemSortFilter	ItemSortFilterCodeType	(eBay Store owners only) Specifies how to sort cross-promotions
UpSellItemFormatSortFilter	ItemFormatSortFilterCodeType	(eBay Store owners only) Specifies what items to show for up-sells.

Table 30-8

Property	Type	Purpose
UpSellGallerySortFilter	GallerySortFilterCodeType	(eBay Store owners only) Specifies Gallery-related options for up-sells.
UpSellItemSortFilter	ItemSortFilterCodeType	(eBay Store owners only) Specifies what items to show for up-sells.

For more information on this call, see “[SetUserPreferences](#)” on page 940. For information on all the properties of the preferences categories that are accepted as input by the call, see the eBay schema documentation (for a link, see “[Schema Reference Documentation](#)” on page 967).

Using the Seller Dashboard to Monitor Your Status

If you are a seller, your Seller Dashboard gives you a quick way to see how you’re doing as an eBay seller. eBay appreciates how hard you work and how important it is to keep track of your business. We designed this tool to help you monitor your progress, maintain your excellent service to our community of buyers, and keep your account in good standing.

We suggest you check your Seller Dashboard regularly to keep track of your performance. For example:

- See if you qualify for improved visibility in Best Match searches with your Search standing.
- Check your Performance Status to see if you are a Top-Rated seller; Top-Rated sellers qualify the highest discounts and eBay services.
- Check your PowerSeller level and see if you qualify for discounts off your Final Value Fees.
- View your Buyer Satisfaction Status to keep track of your customer service rating and see how you can improve your buyers’ experiences.
- Find out if you have any eBay policy violations that might negatively affect your selling, or other situations that require your immediate attention to avoid having restrictions placed on your account.

eBay has a rewards program for sellers who provide excellent customer service. For full details on the rewards programs, review the following eBay Seller Update:

<http://pages.ebay.com/sell/July2009Update/Details/index.html#1-2>

For the latest information about the Seller Dashboard features and policies, see the eBay Web site online help:

About Your Seller Dashboard (eBay US online help)
<http://pages.ebay.com/help/myebay/seller-dashboard.html>

Zum Thema Verkäufer-Cockpit (eBay Germany online help)
<http://pages.ebay.de/help/myebay/seller-dashboard.html>

Note: We are providing links to the eBay US and eBay Germany help as examples. See the corresponding help pages on other eBay sites for similar information.

Use **GetSellerDashboard** to retrieve your Seller Dashboard. See “[GetSellerDashboard](#)” on page 857.

If your application is displaying a Seller Dashboard to a user, it may also be helpful to use **GetFeedback** to provide related information, such as Detailed Seller Ratings. See “[GetFeedback](#)” on page 831.

Getting Account Information

You use **GetAccount** to retrieve the seller account data of an eBay user. **GetAccount** retrieves the seller account data for the user making the request (i.e., the eBay user specified in the credentials of the security header for the call). A call requestor can only retrieve his or her own seller account data. No requestor may retrieve the seller account data for another user.

Seller account data consists of a set of account entries. Each account entry can be either a debit assessed against the account or a credit applied to the account.

GetAccount returns the seller account data in one of three report formats:

- Invoice (either a specific invoice or the last invoice)
- Period
- All History

Indicate which type of report and (if applicable to a particular report type) the date period for the data using properties of the **GetAccount** object.

- **AccountHistorySelection:** Determines which type of report is generated.
- **InvoiceDate:** Used with invoice reports, the value of the **InvoiceDate** property indicates the month for which to generate an invoice report.
- **BeginDate:** Used with period reports, the value of the **BeginDate** property indicates the start of the date range for a period report. Data retrieved for this report type will be that for charges that

occurred (were posted to the account) on or after this date. Used with **EndDate** to establish a finite date range for the returned data. You can retrieve information that is up to 4 months old.

- **EndDate:** Used with period reports, the value of the **EndDate** property indicates the end of the date range for a period report. Data retrieved for this report type will be that for charges that occurred (were posted to the account) on or before this date. Used with **BeginDate** to establish a finite date range for the returned data.

For all report types, the **Pagination** property can be used to control how many account entries are returned per single call. The **Pagination.EntriesPerPage** property controls how many account entries are returned with the current call. An account entry is one **AccountEntryType** object returned in the **AccountEntries** property of the object. The **Pagination.PageNumber** property controls which page of data is returned by the current call.

Generating an Invoice Report

Invoice reports are sets of seller account data that are analogous to the invoices that eBay issues on a monthly basis to sellers. An invoice report contains the account activities and charges for a one-month period. There are two types of invoice reports that can be generated using **GetAccount**: a report for a specified invoice and the last invoice for that seller.

To generate an invoice report for a specific month, set the **AccountHistorySelection** property to **SpecifiedInvoice** and set the **InvoiceDate** property to a date in the month of the target invoice month. (From the date passed in **InvoiceDate**, only the month and year are used to generate the report. Other elements of the date are discarded.)

Note: eBay archives user account data older than four months. As a result, that data is not available for retrieval. Specify an invoice month that is within the past four months.

To generate a report for the last invoice issued to the user by eBay, set the **AccountHistorySelection** property to **LastInvoice**. No value is needed for the **InvoiceDate** property and if one is specified it is ignored.

Do not specify values for the **BeginDate** or **EndDate** properties. They do not apply to this report type.

Generating a Period Report

A period report contains the charges that were applied to the seller's account between two specified dates. Unlike an invoice report, a period report may cover less than one full month or it may cover more than a single month.

To generate a period report, set the **AccountHistorySelection** property to **BetweenSpecifiedDates**. To indicate the dates between which charge dates must fall to be included in the report, specify date values for the **BeginDate** and **EndDate** properties. A particular charge to the seller's account is returned as an account entry (an **AccountEntryType** object in the **AccountEntries** property of the response object) if the date of the charge is on or after the date specified in **BeginDate** and on or before the date specified in **EndDate**.

Note: eBay archives user account data older than four months. As a result, that data is not available for retrieval. Specify a beginning data for the report four months not more than four months prior to the date and time the call is made.

Do not specify a value for the `InvoiceDate` property as it does not apply to this report type.

Working with the GetAccount Result Set

`GetAccount` returns seller account data in the following primary properties of the `GetAccount` object:

- **AccountEntries:** Contains a list of zero, one, or multiple individual account entries.
- **AccountId:** The `AccountId` property (type `string`) contains the unique identifier for the requesting user's seller account.
- **AccountSummary:** Contains summary data for the requesting user's seller account and information on additional accounts.
- **Currency:** The value of the `Currency` property (type `CurrencyCodeType`) indicates the currency in which the currently active account's data is expressed.
- **HasMoreEntries:** Indicates whether there are more account entries yet to be retrieved, considering the number of entries retrieved so far.
- **EntriesPerPage:** Indicates the number of account entries (the `AccountEntryType` objects retrieved in `AccountEntries` property).
- **PaginationResult:** The `PaginationResult` property (type `PaginationResultType`). The child elements of this property indicate the total number of pages of data and the total number of individual account entries there are to be returned.

Accessing Data in the AccountEntries Property

The `AccountEntries` property (which is of type `AccountEntriesType`) contains the individual account entries for the report. Each account entry is represented by an `AccountEntryType` object. Zero, one, or multiple `AccountEntryType` objects may be returned by a single call, depending on how many account entries there are to be returned and how the pagination of data is configured.

Each `AccountEntryType` object contains the data for one account entry. Data returned for an account entry includes the date of the charge or credit, the amount of the entry, an item ID with which the entry is associated (if applicable, such as for the Final Value Fee for an item listing), the balance of the account, a flag indicating the reason for the charge or credit, and more.

Accessing Data in the AccountSummary Property

The `AccountSummary` property (type `AccountSummaryType`) contains the summary data for the account. Summary information includes such account attributes as account state, bank account information, the date of the last invoice to the seller, and the date of the last payment to eBay made by the seller.

The payment types a user may select for making payments to eBay is not the same for all eBay sites. See "[Seller Payment Methods \(SellerPaymentMethod\)](#)" on page 608 for a list of the seller payment methods that are applicable to each eBay site. A user may only select a payment method from those available on the eBay site with which the user is registered. (Seller payment methods are defined in the `SellerPaymentMethodCodeType` code list.)

In addition to summary information for the requesting user's seller account, the [AccountSummary](#) property may contain information for zero, one, or multiple additional accounts. An additional account is one that the user held with a country of residency other than the current one. An eBay user can have only one active account at a time. If a user changes the country of residency, then the new country of residence becomes the currently active account. Any account associated with a previous country of residency is treated as an additional account. Because the currency for these additional accounts are different than the active account, each additional account includes an indicator of the currency for that account. Users who never change their country of residence will not have any additional accounts. Each additional account is represented by an [AdditionalAccount](#) object and the [AccountSummary.AdditionalAccountsCount](#) property indicates how many additional accounts there are.

Uniquely Identifying an Account Entry

You can uniquely identify the entries that are returned within the [AccountEntries](#) tag.

Specifically, the following fields, returned within the [AccountEntries](#) tag, can be used in combination as a unique key: [AccountEntry.Date](#), [AccountEntry.AccountDetailsEntryType](#), [AccountEntry.ItemID](#), and [AccountEntry.RefNumber](#).

Getting Information about a User's Store

You can retrieve information about a user's Store, including the custom categories, by using [GetStore](#). For more information, see "[Editing the Store Settings](#)" on page 536.

Email and Address Privacy Policy

Email Privacy Policy

You cannot retrieve an email address for any user with whom you do not have an order relationship, regardless of site.

Email is only returned for applicable calls when you are retrieving your own user data OR when you and the other user are in an order relationship and the call is being executed within a certain amount of time after the order line item is created. Based on Trust and Safety policies, the time is unspecified and can vary by site.

RegistrationAddress Privacy Policy

eBay always returns complete RegistrationAddress details (including Phone) when you retrieve your own user data. RegistrationAddress for another user (except for Phone) is only returned if you have an order relationship with that user AND that user is registered on DE/AT/CH, regardless of your registration site and the site to which you send the request.

Chapter 31

My eBay

This chapter covers eBay features that handle special user-related features, such as Second Chance Offer, and replicating the functionality of the eBay site's My eBay page in an application.

["My eBay Overview"](#) on page 525

["Replicating My eBay in an Application"](#) on page 526

My eBay Overview

The My eBay page is a more personalized side of the eBay site. Category listing and search results pages are more generalized, showing items offered by many different sellers and with which a viewing user might have no direct connection. The My eBay page shows lists of items that the user is directly associated with—as a buyer or as a seller.

eBay users come to the My eBay page to get summary and specific data about their buying and selling, to set their preferences, and to manage their accounts. 3rd-party applications that help users manage their auctions will often need to provide at least some of the features that My eBay provides.

The content of a user's My eBay page is specific to that user. This content consists of a number of different item lists. These lists range from items on which the user is currently bidding, to items the user is watching, to items the user is currently selling. The lists are spread across a number of tab pages. API

The API provides ways to return data that has the same structure as that found on a user's My eBay page and is filtered to show just those items with which the user is associated. This enables an application to produce a facsimile of the My eBay page.

Replicating My eBay in an Application

Replicating My eBay Views with the API

The My eBay page shows a number of different views of a user's data. This image shows the left navigation menu of the My eBay page that lists the different data views.



Developers building applications that primarily facilitate buying, that is, searching for, watching, monitoring favorites, and bidding on items, will be most interested in the All Buying view.

Developers building applications that primarily facilitate selling, that is, listing items, monitoring paid and shipped status, and relisting, will be most interested in the All Selling view. The My Messages and My Account views apply to all users.

The table below shows the My eBay views that can be retrieved and which API call is used to retrieve the data for a particular section. Each call in the table is hyperlinked to the corresponding Call Reference chapter.

Table 31-1 Replicating My eBay Views with the API

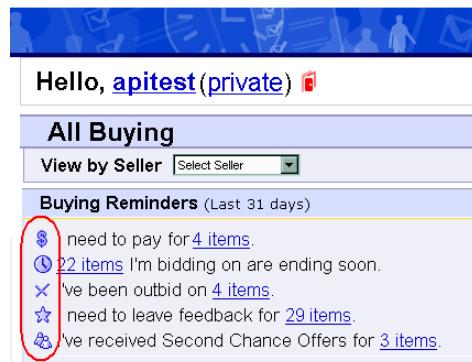
My eBay View	SubItem	Call	Applicable Node
All Buying	Buying Reminders	GetMyeBayReminders	BuyingReminders
	Watching	GetMyeBayBuying	WatchList
	Bidding	GetMyeBayBuying	BidList
	BestOffers	GetMyeBayBuying	BestOfferList
	Won	GetMyeBayBuying	WonList
	Didn't Win	GetMyeBayBuying	LostList
All Selling	Selling Reminders	GetMyeBayReminders	SellingReminders
	Scheduled	GetMyeBaySelling	ScheduledList
	Selling	GetMyeBaySelling	ActiveList
	Sold	GetMyeBaySelling	SoldList
	Unsold	GetMyeBaySelling	UnsoldList

Table 31-1 Replicating My eBay Views with the API (Continued)

My eBay View	SubItem	Call	Applicable Node
My Messages	Read Messages	GetMyMessages	Messages
	Flag	ReviseMyMessages	MessageIDs
	Delete	DeleteMyMessages	MessageIDs
	Revise Folders	ReviseMyMessagesFolders	FolderID
	Reply to ASQ	AddMemberMessageRTQ	MemberMessage
All Favorites	Searches	GetMyeBayBuying	FavoriteSearches
	Sellers	GetMyeBayBuying	FavoriteSellers
My Account	Feedback	GetFeedback	FeedbackDetailArray
	Notifications	SetNotificationPreferences	UserDeliveryPreferenceArray
	Preferences	SetUserPreferences	(all)

Replicating Specific My eBay Views with the API

The icon widgets in each specific My eBay view enable users to see at a glance if there are tasks that require their attention. For example, this image shows the icon widgets displayed in the Buying Reminders view of the My eBay page.



Many widgets have direct API equivalents. The icon widgets and their equivalent API nodes are listed in the table below.

Note: The icon widgets shown below are the property of eBay. You must create your own icons if you plan to use a "widget view" of My eBay in your own application.

See the [eBay eBay Trading API Call Reference](#) for the conditions under which each node is returned. For convenience, each API Equivalent node is hyperlinked to the corresponding element in the [eBay eBay Trading API Call Reference](#).

Table 31-2 Replicating Specific My eBay Views with the API

Widget	Meaning	API Equivalent
All Buying: Buying Reminders (organized by totals)		

Table 31-2 Replicating Specific My eBay Views with the API (Continued)

Widget	Meaning	API Equivalent
	■ I need to pay for...	■ <code>GetMyeBayRemindersResponse.BuyingReminders.PaymentToSendCount</code>
	■ I've been outbid on...	■ <code>GetMyeBayRemindersResponse.BuyingReminders.OutbidCount</code> Indicates the number of items that the user has been outbid on. See " GetMyeBayBuying " on page 839 for a sample that shows how to programmatically get a list of items that a bidder is winning or has been outbid on.
	■ I need to leave feedback for...	■ <code>GetMyeBayRemindersResponse.BuyingReminders.FeedbackToSendCount</code>
	■ I've received Second Chance Offers for...	■ <code>GetMyeBayBuyingResponse.SecondChanceOffer</code>
All Buying: Items I've Won (organized by item)		
	■ Item needs to be paid	■ <code>GetMyeBayBuyingResponse.WonList.OrderTransactionArray.OrderTransaction.Order.TransactionArray.Transaction.BuyerPaidStatus</code> A <code>BuyerPaidStatus</code> of <code>NotPaid</code> indicates that the item still needs to be paid for. Note that some buyer payment methods, such as cash, cannot be tracked by eBay, so the buyer or seller must manually mark the item <code>MarkedAsPaid</code> to have it show up in My eBay.
	■ Feedback needs to be left	■ <code>GetMyeBayBuyingResponse.WonList.OrderTransactionArray.OrderTransaction.Order.TransactionArray.Transaction.FeedbackLeft</code> If <code>FeedbackLeft</code> is absent from the response, feedback still needs to be left for the item. Use <code>LeaveFeedback</code> to leave feedback.
	■ Feedback not yet received	■ <code>GetMyeBayBuyingResponse.WonList.OrderTransactionArray.OrderTransaction.Order.TransactionArray.Transaction.FeedbackReceived</code> If <code>FeedbackReceived</code> is absent from the response, feedback has not yet been received. Use <code>LeaveFeedback</code> to leave feedback.
All Selling: Selling Reminders (organized by totals)		
	■ I am awaiting payment for...	■ <code>GetMyeBayRemindersResponse.SellingReminders.PaymentToReceiveCount</code>
	■ I need to leave feedback for...	■ <code>GetMyeBayRemindersResponse.SellingReminders.FeedbackToSendCount</code>
All Selling: Items I've Sold (organized by item)		
	■ Feedback not left	■ <code>GetMyeBaySellingResponse.SoldList.OrderTransactionArray.OrderTransaction.Order.TransactionArray.Transaction.FeedbackLeft</code> If <code>FeedbackLeft</code> is absent from the response, feedback still needs to be left for the item. Use <code>LeaveFeedback</code> to leave feedback.
	■ Feedback left	■ <code>GetMyeBaySellingResponse.SoldList.OrderTransactionArray.OrderTransaction.Order.TransactionArray.Transaction.FeedbackLeft</code> If <code>FeedbackLeft</code> is present in the response, feedback has been left for the item.

Table 31-2 Replicating Specific My eBay Views with the API (Continued)

Widget	Meaning	API Equivalent
	■ Feedback not yet received	<ul style="list-style-type: none"> ■ <code>GetMyeBaySellingResponse.SoldList.OrderTransactionArray.OrderTransaction.Order.TransactionArray.Transaction.FeedbackReceived</code> <p>If <code>FeedbackReceived</code> is absent from the response, feedback has not yet been received. Send a message requesting feedback using <code>AddMemberMessageAAQToPartner</code>.</p>
	■ Feedback received	<ul style="list-style-type: none"> ■ <code>GetMyeBaySellingResponse.SoldList.OrderTransactionArray.OrderTransaction.Order.TransactionArray.Transaction.FeedbackReceived</code> <p>If <code>FeedbackReceived</code> is present in the response, feedback has been received. <code>FeedbackReceived.CommentType</code> indicates whether the feedback was positive, negative, or neutral. Use <code>LeaveFeedback</code> or <code>CompleteSale</code> to leave feedback in response.</p>
My Messages		
	■ I have new messages...	<ul style="list-style-type: none"> ■ Value displayed in My eBay is the total of <code>GetMyMessagesResponse.Summary.NewMessageCount</code> <p>Use <code>GetMyMessages</code> to retrieve new messages.</p>
	■ I have flagged messages...	<ul style="list-style-type: none"> ■ <code>GetMyMessagesResponse.Messages.Message.Flagged</code> <p>Indicates the number of messages that are flagged. Use <code>GetMyMessages</code> to retrieve flagged messages, and <code>ReviseMyMessages</code> to flag and unflag messages.</p>

Retrieving Items with GetMyeBayBuying

In the schema, a request to `GetMyeBayBuying` allows you to specify which nodes you want returned in the response and customize the response list. The specific elements you can customize in the list vary by the node you want returned (see the schema documentation for details).

If you want to return a `WatchList` node, to represent the items a user is watching, you can customize the request to `GetMyeBayBuying` like this.

Example 31-1 Customizing the Return List of GetMyeBayBuying (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<GetMyeBayBuyingRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <Version>425</Version>
  <DetailLevel>ReturnAll</DetailLevel>
  <WatchList>
    <Sort>TimeLeft</Sort>
    <IncludeNotes>true</IncludeNotes>
    <Pagination>
      <EntriesPerPage>10</EntriesPerPage>
      <PageNumber>1</PageNumber>
    </Pagination>
  </WatchList>
</GetMyeBayBuyingRequest>
```

In the `GetMyeBayBuying` response, the nodes `WatchList`, `BidList`, `WonList`, and `LostList` are based on the complex type `ItemListCustomizationType`, which gives you pagination through the `Pagination` element. It is recommended that you use pagination, as the result list can be quite long.

The **WatchList** returned by **GetMyeBayBuying** is an **ItemArray** that looks like [Example 31-2](#).

Example 31-2 Watch List Returned by GetMyeBayBuying (XML)

```
<WatchList>
  <ItemArray>
    <Item>
      <BuyItNowPrice currencyID="USD">4.0</BuyItNowPrice>
      <ItemID>705058049</ItemID>
      <ListingDetails>
        <StartTime>2005-08-08T17:29:07.000Z</StartTime>
        <EndTime>2005-08-09T04:29:07.000Z</EndTime>
      </ListingDetails>
      <Seller>
        <UserID>abe7</UserID>
      </Seller>
      <SellingStatus>
        <CurrentPrice currencyID="USD">4.0</CurrentPrice>
      </SellingStatus>
      <TimeLeft>PT0S</TimeLeft>
      <Title>Test Domains</Title>
      <QuantityAvailable>4</QuantityAvailable>
    </Item>
    ...
  </ItemArray>
</WatchList>
<PaginationResult>
  <TotalNumberOfPages>1</TotalNumberOfPages>
  <TotalNumberOfEntries>5</TotalNumberOfEntries>
</PaginationResult>
</WatchList>
```

Most of the other nodes **GetMyeBayBuying** returns work in a similar way, allowing you to customize the node on the request and returning an array. However, **FavoriteSearches** and **FavoriteSellers** simply return a group of nodes, without structuring them into an array, as shown in [Example 31-3](#).

The **SearchQuery** value is a HTML-encoded URL string that many applications can use directly.

Example 31-3 Favorite Searches and Sellers Returned by GetMyeBayBuying

```

<FavoriteSearches>
  <TotalAvailable>3</TotalAvailable>
  <FavoriteSearch>
    <SearchName>Database systems</SearchName>
    <SearchQuery>http://search-desc.qa.ebay.com/ws/search/
SaleSearch?basicsearch=&from=R10&ht=1&sacategory=267&saregion=0&satitle=Database+
systems&sorecordsperpage=50&sosortorder=1&sosortproperty=1&sotextsearched=2</
SearchQuery>
  </FavoriteSearch>
  <FavoriteSearch>
    <SearchName>oreilly</SearchName>
    <SearchQuery>http://search-desc.qa.ebay.com/ws/search/
SaleSearch?basicsearch=&from=R10&ht=1&sacategory=267&saregion=0&satitle=oreilly&sorecordsperpage
=50&sosortorder=1&sosortproperty=1&sotextsearched=2</SearchQuery>
  </FavoriteSearch>
  <FavoriteSearch>
    <SearchName>Unnamed Search</SearchName>
    <SearchQuery>http://motors.search.qa.ebay.com/ws/search/
SaleSearch?catref=C6&fisc=c6024&from=R9&sacategory=6024&samet=2&socolumnlayout=3&
amp;sofindtype=22&sorecordsperpage=50&sosortorder=1&sosortproperty=1&sotrtype=1&s
otrvalue=1</SearchQuery>
  </FavoriteSearch>
</FavoriteSearches>
<FavoriteSellers>
  <TotalAvailable>2</TotalAvailable>
  <FavoriteSeller>
    <UserID>sampleuser</UserID>
    <StoreName>panache_store</StoreName>
  </FavoriteSeller>
  <FavoriteSeller>
    <UserID>sampleuser</UserID>
    <StoreName>eWatch Crew Testing Store</StoreName>
  </FavoriteSeller>
</FavoriteSellers>

```

Retrieving Items with GetMyeBaySelling

In the schema, the request to [GetMyeBaySelling](#) is similar to the request to [GetMyeBayBuying](#). You can specify values in the request to customize the return list.

Example 31-4 Customizing the Return List of GetMyeBaySelling

```

<?xml version="1.0" encoding="UTF-8"?>
<GetMyeBaySellingRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <Version>425</Version>
  <ActiveList>
    <Sort>TimeLeft</Sort>
    <Pagination>
      <EntriesPerPage>10</EntriesPerPage>
      <PageNumber>1</PageNumber>
    </Pagination>
  </ActiveList>
</GetMyeBaySellingRequest>

```

This request is for the **ActiveList**, so the response looks like [Example 31-5](#).

Example 31-5 ActiveList Returned by GetMyeBaySelling

```
<ActiveList>
  <ItemArray>
    <Item>
      <ItemID>9203288303</ItemID>
      <ListingDetails>
        <ConvertedStartPrice currencyID="USD">2.89</ConvertedStartPrice>
        <ConvertedReservePrice currencyID="USD">0.0</ConvertedReservePrice>
        <StartTime>2005-07-21T18:41:04.000Z</StartTime>
      </ListingDetails>
      <ListingType>Chinese</ListingType>
      <Quantity>1</Quantity>
      <ReservePrice currencyID="MYR">0.0</ReservePrice>
      <SellingStatus>
        <ConvertedCurrentPrice currencyID="USD">2.89</ConvertedCurrentPrice>
        <CurrentPrice currencyID="MYR">11.0</CurrentPrice>
        <ReserveMet>true</ReserveMet>
      </SellingStatus>
      <StartPrice currencyID="MYR">11.0</StartPrice>
      <TimeLeft>PT28M17S</TimeLeft>
      <Title>Test Auction Title</Title>
      <QuantityAvailable>1</QuantityAvailable>
    </Item>
    ... more Item nodes ...
  </ItemArray>
  <PaginationResult>
    <TotalNumberOfPages>1147</TotalNumberOfPages>
    <TotalNumberOfEntries>3440</TotalNumberOfEntries>
  </PaginationResult>
</ActiveList>
```



Part VIII

eBay Stores

Managing eBay Stores	534
Cross-Promotions	544

Chapter 32

Managing eBay Stores

This chapter discusses using the API to manage an eBay Store.

eBay Stores is the premium merchandising space on eBay. A seller with an eBay Store (*eBay Stores sellers*) can showcase their listings in one customized, searchable location. They can show all of their items together and tell buyers more about their business on their own customized pages. When a buyer goes to the seller's store, they can choose to search within that seller's items only. A Stores seller can list in multiple quantities for longer durations and create custom categories (which are like store departments). Plus, sellers with an eBay Store have advanced merchandising tools that allow them to control which of their items are displayed on key buyer pages.

Sellers with an eBay Store get a powerful set of features to help them sell more on eBay. Plus, Stores sellers get their own distinctive location to showcase all their eBay listings with their own customized storefront and custom Store URL.

When a user who happens to be an eBay Stores seller lists auction-style and fixed price listings on eBay, those listings also appear in the seller's eBay Store.

Each distinct Stores purchase is assigned a unique identifier, or transaction ID. This transaction ID is used for the operations that follow the purchase, including payment and feedback.

For more information on eBay Stores, please see the Store Benefits page on the eBay Web site.

Learn More about the Benefits of eBay Stores
<http://pages.ebay.com/storefronts/openbenefits.html>

Sellers can create, edit, and manage their Stores on the eBay website using the Store Builder tool, while the API allows sellers to edit and manage their eBay Store.

The following sections discuss various aspects of editing and managing an eBay Store:

- “[Editing the Store Settings](#)” on page 536
- “[Managing Store Categories](#)” on page 539
- “[Managing Custom Pages](#)” on page 540
- “[Managing a User’s Store Preferences](#)” on page 542
- “[Special Listing Fields for eBay Stores Sellers](#)” on page 543

Creating a Store

These instructions explain how to use an online form to create eBay Stores in the Production environment and in the Sandbox.

In the **Production** environment, a user is eligible to create an eBay Store if he or she is a registered seller on eBay and meets one of the following conditions:

- has a cumulative feedback score of at least 20
- is registered with ID Verify
- has a PayPal account in good standing

In the **Sandbox**, to be eligible to create an eBay Store:

- You need to use a Sandbox test user (not your normal eBay username, not your eBay Developers Program username).
For information about creating test users, see “[Creating a Test User](#)” on page 63.
- The test user must first list at least one item for sale in the Sandbox.
- The test user must have a PayPal Sandbox account linked to that user’s eBay Sandbox account. See “[Setting Up the PayPal Sandbox](#)” on page 199.

Important: A known issue may prevent you from linking your eBay Sandbox and PayPal Sandbox accounts yourself. This means you may have difficulty creating a store in the eBay Sandbox. Please check the Sandbox System Announcements for more information and workarounds.

Sandbox System Announcement

PayPal Linking is not working - Will not be able to create a store

<http://dev-forums.ebay.com/thread.jspa?threadID=500017502&tstart=0&mod=1294879751434>

To set up an eBay Store:

- 1 Using a browser, go to the eBay Stores site:

eBay Stores (US) in Production
<http://stores.ebay.com>

eBay Stores in the Sandbox
<http://stores.sandbox.ebay.com/>

- 2 Click the **Open an eBay Store** link in the upper right-hand corner of the eBay Stores page.
- 3 Sign in and follow the instructions to create the user’s eBay Store. (If you are creating a Store in the **Sandbox**, be sure to sign in as the test user who will be the store owner.)

Once you have set up an eBay Store, you can edit and manage it programmatically through the API .

Resources of Interest Related to eBay Stores

Refer to “[Working with Features for eBay Stores Sellers](#)” on page 557 for an overview of eBay store features, eBay Motors items in eBay Stores, and special fields for eBay Stores sellers. The URLs below point to online help resources on the eBay site. They can assist you in understanding how Stores operate, as well as the rules and requirements governing listing items in eBay Stores.

[Listing in Your Store](#)

<http://pages.ebay.com/storefronts/seller-landing.html>

[Frequently Asked Seller Questions About eBay Stores](#)

<http://pages.ebay.com/storefronts/faq.html>

Editing the Store Settings

eBay Store owners can customize the look and feel of a store in two ways:

- By choosing a predefined theme
- By specifying the exact colors and graphics

In addition to specifying the look and feel of the Store, you can specify such things as how items are to be listed and a custom header and header style for the Store. No matter what change you want to make to the Store configuration using the API, you follow the same general procedure.

Step 1: Retrieve the List of Store Options

Some of the options for eBay Stores can change frequently. For this reason, these options are not specified in the Schema, since this would require users to update their Schema every time an option changed. You can obtain the current list of options for eBay Stores by calling [GetStoreOptions](#). These are the types of options that can be retrieved with this call:

- Themes (basic and advanced)
- Logos
- Color schemes
- Subscription names and rates

You should make this call before making changes to a Store configuration to ensure that you are using the most recent set of options.

The returned list of options splits the themes into basic (in the [BasicThemeArray](#) property) and advanced (in the [AdvancedThemeArray](#) property). Each [BasicThemeArray](#).[Theme](#) property contains a color scheme that is compatible with it in the [BasicThemeArray](#).[Theme](#).[ColorScheme](#).[ColorSchemeID](#) property. You will notice that the [ThemeID](#) properties are repeated; each basic theme has a number of color schemes that are compatible with it. The advanced themes do not contain color schemes, because you can use any [AdvancedThemeArray](#).[Theme](#).[ThemeID](#) with any of the generic color scheme IDs in [AdvancedThemeArray](#).[GenericColorSchemeArray](#).[ColorScheme](#).[ColorSchemeID](#).

You can get a user-friendly name for the themes, color schemes, and logos that are returned by this call:

- `BasicThemeArray.Theme.Name`
- `BasicThemeArray.Theme.ColorScheme.Name`
- `AdvancedThemeArray.Theme.Name`
- `AdvancedThemeArray.GenericColorSchemeArray.ColorScheme.Name`
- `LogoArray.Logo.Name`

Step 2: Retrieve the Current Store Configuration

Before making any changes to a Store configuration, you may want to display the current configuration for the Store. You obtain the configuration by making a call to [GetStore](#).

You can reuse the `Store` object that you retrieve from the call to [GetStore](#) if you choose to make a change to the Store configuration in [SetStore](#). Simply change the values of one or more properties of the object before submitting it make the changes.

Step 3: Submit the Modified Store Configuration

You make changes to the Store configuration by calling [SetStore](#). You submit a `Store` object to this call, which contains all the properties that specify the configuration.

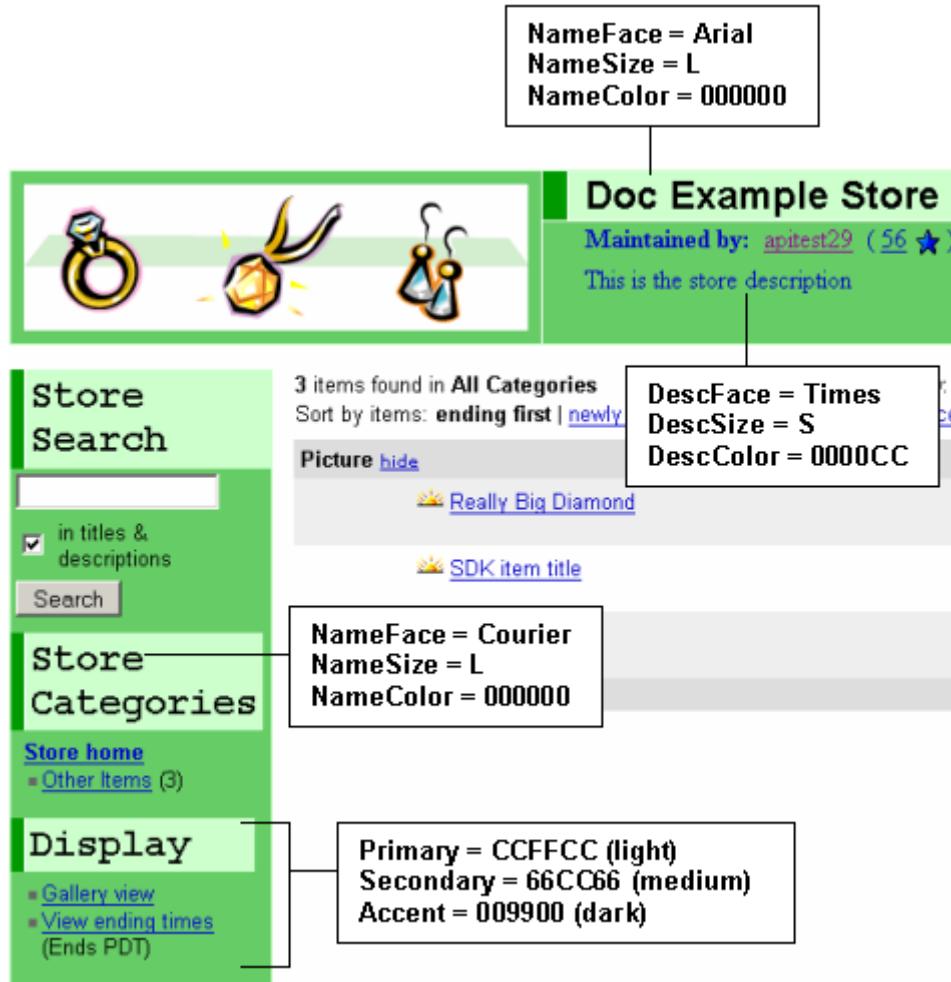
Refer to the [SetStore](#) Input section of the [eBay Trading API Call Reference](#) for descriptions of the [SetStore](#) input fields for configuring an eBay Store.

When you are configuring the look and feel of a Store, you can choose between picking a pre-defined theme and color scheme, or manually filling in the properties that define the look and feel. Following are the rules for how the configuration when various properties of the **Store** object are specified:

- If **ThemelD** is specified, and it is different from the existing **ThemelD** for the **Store**, then the **ColorSchemeID** should be specified too, and vice-versa.
- If a **ThemelD** and **ColorSchemeID** are specified, and they are same as the ones already on record (the **ThemelD** and **ColorSchemeID** are not being changed), then one of the following occurs:
 - If you specify one or more of the 12 look and feel settings (colors, fonts, etc...), they will replace the existing values on record (the values will be set to the defaults if you are editing the theme for the first time).
 - If none of these 12 settings are specified, nothing changes.
- If **ThemelD** and **ColorSchemeID** are specified, and they are different from the ones on record (the **ThemelD** and **ColorSchemeID** are being changed), then one of the following occurs:
 - First the **ThemelD** and **ColorSchemeID** combination is validated (Is the **ThemelD** correct? Is the **ColorSchemeID** correct? Is the **ColorSchemeID** valid for this **ThemelD**?)
 - If you specify one or more of the 12 look and feel settings, then they will be saved in the database. The missing ones are populated with default values.
 - If you don't specify any of those 12 settings, all the 12 settings will have default settings.

All properties of the **Store** object are optional, so you only need fill in those properties with those values that you wish to change.

The following figure shows how the custom look and feel properties are used on the Store page:



Note that there is no built-in capability in the API to preview how the modified Store will appear. Once changes to a Store configuration are submitted, you can view the URL for the Store to see how it looks.

Managing Store Categories

An eBay Store owner can use [SetStoreCategories](#) to modify the category structure of his or her eBay Store. Store categories can be added, deleted, moved, and renamed with [SetStoreCategories](#). Changes can be made to a single store category or to a list of store categories. Using the API, you can add up to 300 store categories to an eBay Store. Stores support three levels of categories. Use [GetStore](#) to retrieve the category structure for your store.

Note: Store categories will not appear in the site user interface unless they contain items.

Items can only be contained within child store categories. A parent category cannot contain items. If adding, moving, or deleting categories displaces items, you must use the [ItemDestinationCategoryId](#) input field to specify a destination child category under which the displaced items will be moved. The destination category must have no child categories.

Refer to the [SetStoreCategories](#) section of the [eBay Trading API Call Reference](#) for descriptions of the [SetStoreCategories](#) input and output fields. For more information on [SetStoreCategories](#), including samples of the XML API call structures, please see “[SetStoreCategories](#)” on page 935.

Managing Custom Pages

Every Store can optionally include a number of custom pages. These pages can include static HTML content as well as dynamic content that is added using the special eBay Stores tags. The number of custom pages allowed for a Store depends on the subscription level.

Examples of custom pages include the following:

- Frequently Asked Questions (FAQ) page
- Custom storefront page
- General information about one or more items for sale in a Store

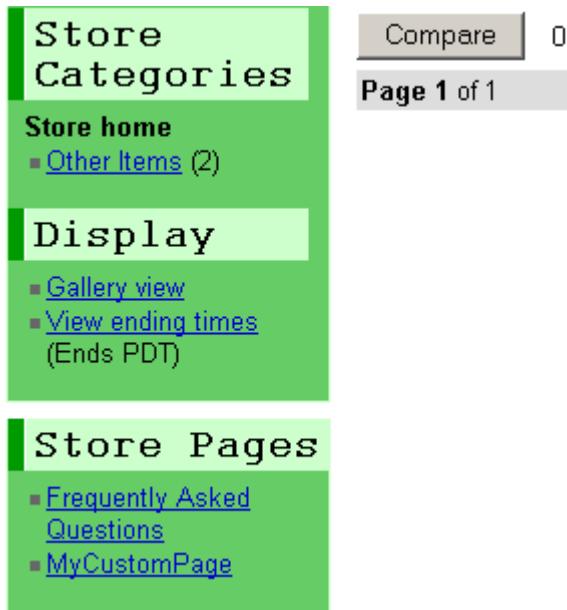
You can use the API to add or delete custom pages. You can also mark pages as inactive so that they are not visible to users.

Step 1: Getting the Custom Page List

Each eBay Store contains a list of its custom pages. Unless you already know which custom page you want to edit, then the first thing you must do is retrieve the current list of the Store’s custom pages. To retrieve the list, make a call to [GetStoreCustomPage](#) without specifying any specific [PageID](#). The call returns information about all the custom pages, including the [PageID](#), [Name](#), and the [Order](#) in which the page appears.

Refer to the [GetStoreCustomPage](#) section of the [eBay Trading API Call Reference](#) for descriptions of the [GetStoreCustomPage](#) input and output fields.

Links to a Store's custom pages appear on the bottom left of the Store's navigation bar:



The **Name** of the custom page is used in links to the page, and the **Order** property is used to determine the order of the pages in the list of links.

Step 2: Getting a Single Custom Page

Once you have retrieved the list of custom pages for a Store, you can get the whole content of a single page by making a call to [GetStoreCustomPage](#), specifying the **PageID** for which page to return.

The return value contains all the same properties for the page you are requesting as were returned when calling [GetStoreCustomPage](#) with no **PageID**, plus a few more. The most significant property returned is **CustomPageArray.CustomPage.Content**. This property contains the HTML content of the custom page.

For more information on [GetStoreCustomPage](#), please see “[GetStoreCustomPage](#)” on page 870.

Step 3: Submitting a Modified or New Custom Page

You can either modify the custom page that you have retrieved, or you can create a new one if you are starting from scratch. Once you have created the HTML content of the custom page, submit the changes by using [SetStoreCustomPage](#). Refer to the [SetStoreCustomPage](#) Input section of the [eBay Trading API Call Reference](#) for descriptions of the [SetStoreCustomPage](#) input fields.

eBay provides a set of special HTML tags for use in Store custom pages and headers. You can use these tags to do such things as display your feedback rating, a list of items for sale, or a particular item. To use these special HTML tags in your custom page, simply include them in the content of your page which you specify in the **CustomPage.Content** property. For more information on these special tags, please see the following page in the eBay Help:

<http://pages.ebay.com/help/stores/stores-specific-tags.html>

The custom page preview feature allows you to view how the page will appear to buyers before the page is added or modified. This is especially useful when you use any of the special Stores HTML tags in your custom page, so that you can see how the page appears when eBay processes the special tags. To use the preview feature, specify the `CustomPage.PreviewEnabled` property as `true`. To add or modify the custom page after you have previewed it, make a call to `SetStoreCustomPage` with either the same or with different inputs, specifying the `CustomPage.PreviewEnabled` property as `false`.

For more information on `SetStoreCustomPage`, please see “[SetStoreCustomPage](#)” on page 936.

Managing a User's Store Preferences

You can use the API to manage a user's eBay Store preferences. Such preferences include the ability for the Store owner to indicate that they are currently away on vacation, and options on how to handle current listings during a vacation, such as by hiding fixed price items for the duration.

Setting Store Preferences

You set Store preferences by using `SetStorePreferences`. You specify the preferences that are being set in the `StorePreferences` property of the request object. This property contains one sub-property, `StoreVacationPreferences`. Refer to the `SetStorePreferences` Input section of the [eBay Trading API Call Reference](#) for descriptions of the `SetStorePreferences` input fields.

For more information on the call, see “[SetStorePreferences](#)” on page 937.

When you set the Store preferences, if you specify in `OnVacation` that the Store owner is on vacation, then you must specify one or more of `HideFixedPriceStoreItems`, `MessageItem`, or `MessageStore` as `true`.

For all item types, if a `ReturnDate` is specified and the `MessageItem` property is `true` then the `Item.SellerVacationNote` property on the `GetItem` response indicates that the Store owner will return on the specified date.

After a `ReturnDate` has been set, the only way to remove it is to set the date to `1900-1-1`. If the `OnVacation` property is set to this value when the preferences are submitted, the system will recognize this value and make the field null. Note that if a `ReturnDate` is specified, the system will not change the `OnVacation` property to `false` at that time. The `OnVacation` property must always be explicitly changed by the user.

If the `OnVacation` property is `true` and the `HideFixedPriceStoreItems` property is also `true`, then calling `GetCrossPromotions` will not return listings for the seller.

Getting Store Preferences

You can get the current set of Store preferences for a user by using `GetStorePreferences`. This call returns the same set of preferences as can be set using `SetStorePreferences`. For more information on this call, see “[GetStorePreferences](#)” on page 872.

Special Listing Fields for eBay Stores Sellers

An eBay Stores seller can create up to three levels of custom categories within their Store (either using the eBay Web site or the API) and up to 300 custom categories can be added. Custom categories are often given names that sound like store departments. When executing **AddItem** (or any call in the **AddItem** family of calls) the seller can specify one of these custom categories to place the item within that store category. The custom category information is only applicable when the listing is viewed in the seller's Store. For example, if the listing is for a basic Fixed Price item, the store category won't be applicable when the listing is viewed on the main eBay site. **GetItem** will also return this category ID information.

If a user is an eBay Stores seller, additional fields are returned within the seller role information.

Table 32-1 User Information For eBay Stores Sellers

Item Property	Purpose
<code>Item.Storefront.StoreCategoryID</code>	Enables a seller to associate a listing with one of up to 300 custom categories they have previously set up in their eBay Store. Each custom category has a unique ID. A value of 0 indicates that the item is not an eBay Store item. A value of 1 is used for "Other". Applicable for input and output.
<code>User.SellerBusinessType.Commercial</code>	Indicates whether the specified user is an eBay Stores seller. Output only. (Only returned for sites that have implemented Seller Business Types.)
<code>User.SellerInfo.StoreOwner</code>	Indicates whether the specified user is an eBay Stores seller. Output only.
<code>User.SellerInfo.StoreURL</code>	URL pointing to the user's eBay Store page, if any. Output only.
<code>Item.Seller.SellerInfo.StoreOwner</code>	Indicates whether the specified user is an eBay Stores seller. Output only.
<code>Item.Seller.SellerInfo.StoreURL</code>	URL pointing to the user's eBay Store page, if any. Output only.
<code>Item.SellingStatus.HighBidder.SellerInfo.StoreOwner</code>	Indicates whether the specified user is an eBay Stores seller. Output only.
<code>Item.SellingStatus.HighBidder.SellerInfo.StoreURL</code>	URL pointing to the user's eBay Store page, if any. Output only.

Other calls provide no specialized features for eBay Stores sellers. That is, they treat eBay Store sellers' listings just like listings by sellers who are not Store sellers.

Listing US eBay Motors Items in eBay Stores

Items that an eBay Stores seller lists to the US eBay Motors site (site ID of `eBayMotors`) also appear in the seller's eBay Store.

eBay Motors items within the Parts & Accessories and the Everything Else categories can be listed in auction-style or fixed-price formats. eBay Motors items within vehicle categories can be listed in either auction-style or fixed-price formats.

Chapter 33

Cross-Promotions

eBay Store Cross-Promotions are no longer supported in the Trading API. This chapter may be useful for developers who are still removing support for eBay Store promotion rules. This chapter will be removed soon from the documentation.

[“Overview” on page 544](#)

[“Getting Set Up to Use Merchandising Manager” on page 545](#)

[“Cross-Promotion Methods” on page 545](#)

[“Retrieving Cross-Promotion Rules” on page 554](#)

Overview

When a buyer wins an auction or purchases an item outright, Merchandising Manager allows a seller to cross-promote related items. The seller defines rules that determine which items are cross-promoted, either individually or as a category in the seller's eBay Store.

The cross-promotions appear on various pages on the eBay site, such as the **View Item** page for listed items. Cross-promotions draw the viewer to a related item, showing a page for that item with a thumbnail image and its current price. When your application retrieves item data to display, it also retrieves data for cross-promoted items so that they can be displayed with the original item. The display format is called a widget.

Figure 33-1 Cross-Promoted Items in a Widget



Cross-promoting items with Merchandising Manager brings more of the seller's items to the attention of buyers and bidders. The buyer might purchase more items from the seller. Other benefits gained are not necessarily immediate. For instance, by publicizing other related items to a buyer, the

seller influences that buyer to return to the eBay Store for future purchasing needs—simply by letting that buyer know of items that are available and of interest.

- All of the functionality described here may be used either on the eBay site or through the API with [GetPromotionRules](#) and [GetCrossPromotions](#).

Note: To use the Merchandising Manager and set up promotions, you must be the owner of an eBay Stores storefront, with any level of eBay Stores subscription. This section is written for those sellers.

Getting Set Up to Use Merchandising Manager

To use the Merchandising Manager, you must first set cross-promotion preferences for your eBay store (with [SetUserPreferences](#)) and then define the cross-promotion rules that determine how items are cross-promoted (with Merchandising Manager).

Once you have enabled Merchandising Manager, you should define the cross-promotion rule schemes and the promotion methods that facilitate the cross-promotion. You can set up rule schemes and promotion methods on the eBay site by using the Merchandising Manager interface. You can also retrieve the promotion rules for your store with [GetPromotionRules](#) or the cross-promotions for one item with [GetCrossPromotions](#).

If you opt to participate in Merchandising Manager, but you have not defined any rules, then a default rule may apply. In placement locations where the promotion method is cross-sell, eBay uses a default algorithm to populate the cross-promotion widget with items. For placement locations where the promotion method is upsell, there is no default method for delivering cross-promotion items and the widget does not show up at all.

When you create a cross-promotion, you need to consider four basic aspects:

- **Methods:** The types of cross-promotions you can perform.
- **Rule Schemes:** The rules that decide which items are displayed as cross-promotion items.
- **Placement:** The locations on the eBay site where cross-promotions appear.
- **Presentation Styles:** The graphical manner in which cross-promoted items are presented.

Please note that the item needs to be listed with a Gallery image for the image to show up in the widget.

Cross-Promotion Methods

There are two types of item cross-promotions that can be used with the eBay API: upsell and cross-sell. In an upsell promotion, the seller presents items that are more expensive or higher quality than the item the buyer is bidding on or has purchased. In a cross-sell promotion, the seller presents items that the buyer is likely to find of interest. This is based on items the user has already bought or has bid on. Typically, the cross-sell promotion item is complementary to the item already purchased.

The upsell and cross-sell promotion methods apply to all three of the rule scheme types: **item-to-item**, **item-to-Store-category**, and **Store-category-to-Store-category**. Any one rule can be defined as either an upsell or a cross-sell, depending on how the seller wishes to promote the items.

However, not all cross-promotion methods are applicable to all placement locations. [Table 33-1](#) shows which promotion methods are applicable to which placement locations.

Table 33-1

Placement	Method
Active ViewItem Page (<i>where the buyer confirms the bid</i>)	Cross-sell
ViewItem Page (<i>where the buyer confirms a Buy It Now purchase</i>)	Cross-sell
Closed ViewItem Page (<i>visible to the winning bidder or buyer</i>)	Cross-sell
Closed ViewItem Page (<i>visible to a non-winning bidder, non-participant, or unknown user</i>)	Upsell
ViewItem Page (<i>visible to prospective bidders or buyers</i>)	Upsell

What this means is that a particular cross-promotion item might not be displayed on one placement location, although it might be included on another placement location of a different type based on the promotion method. For example, assume an item-to-item rule defined with the upsell method. None of the specified items would appear on the Active ViewItem Page when the referring item is viewed. But if the referring item is viewed through the Closed ViewItem Page (after the listing has ended), then these items would be used for cross-promotions.

For this reason, a seller might choose to define multiple overlapping rules, one with the upsell method and the other with the cross-sell method to ensure that items are available for cross-promotion no matter what page the user views the referring item through. Naturally, the items cross-promoted would be different for each of the promotion methods as the context for the cross-promotion is different.

Cross-Promotion Rule Schemes

A rule scheme is the means by which a seller defines the merchandising relationship between a referring item (or eBay Stores category) and cross-promotion items. Rule schemes determine which items are displayed to the target user as cross-promotion items. Some rules apply only to one individual referring item, expiring when the item's listing ends. One rule applies to any and all items in a Store category, having effect any time one of that category's items are displayed.

There are three rule schemes that a seller can define using the API:

- **Item-to-item:** The seller specifies individual items (by their item IDs) that are to be displayed as cross-promotion items. When the referring item is viewed, these items are displayed with a link to the promoted item. Also displayed are the item's current price and the type of offer that can be

made. In an API application, this data is retrieved with the item (or separately with a call to [GetCrossPromotions](#)).

- **Item-to-Store-category:** The seller specifies a category in the same eBay Store from which items are drawn and displayed as cross-promotion items.
- **Store-category-to-Store-category:** The seller specifies a category in the same eBay Store as the referring category. When any item from that category is displayed, the rule causes items from the promoted category to be displayed as cross-promotion items.

Query Filters

When a rule scheme is activated, a query is run against the eBay database to retrieve the items that are cross-promoted. This query is subject to a set of logical filters. The criteria for this filtering are listed below.

Note that because of this filtering, it is possible that not all of the items that are present might be used in cross-promotions. For example, in an item-to-Store-category rule, the referring item might be in the promoted category. In this case, the item would fail the first criterion below and would not appear as a cross-promotion item. These filtering criteria apply to all three of the above rule schemes:

- The cross-promotion item cannot be the same as the referring item.
- The cross-promotion item must be an active listing.
- The cross-promotion item cannot be a pending listing.
- The cross-promotion item cannot be a Second Chance Offer listing.
- The cross-promotion item cannot be listed in the Mature Audiences category.
- The cross-promotion item cannot be banned based on the site ID and browser language of the target user.

An item-to-item rule allows the seller to specify for a particular referring listing the exact items that should appear as cross-promotion items. The seller can specify zero to four items. (Also see “[Cross-Promotion Presentation Styles](#)” on page 551, where these four items are associated with four cross-promotion positions in the display.) When the referring item is displayed, the cross-promotion items are not necessarily all of the items specified. The actual items used are those that were not removed from the set by the query filters. The four cross-promotion items are returned in the query in the order prescribed by the seller. This order is used to rank them and position them in the cross-promotion display widget on the eBay site.

If cross-promotion rules are being returned by [GetCrossPromotions](#), this is the order in which the items are returned. Any type of listing (competitive-bidding or fixed-price) belonging to the seller may be designated as a cross-promotion item in an item-to-item rule. There are also no restrictions as to the site on which the cross-promotion item was listed—it can be on the same site as or a different site than the referring item. Should a cross-promotion item not be available when the referring item is displayed (such as when the cross-promotion item is ended early), then another item is selected to take its place in the cross-promotion widget based on the existing item-to-Store-category rule.

Between the time that a seller defines an item-to-item rule and a viewer sees the cross-promotions, it is possible that one or more of the cross-promoted item listings could have ended. To avoid having fewer than four cross-promotion items appear (or none at all) a seller can also define an item-to-Store-category or Store-category-to-Store-category rule scheme. The items from this supplementary (or backfill) rule fill the remainder of cross-promotion positions that would otherwise have been empty. This can be critically important as the cross-promotion display widget does not appear on the placement location unless there are two or more items available for cross-promotion.

Types of Rules

Item-to-item. An item-to-item rule is applicable only to an individual listing (because it requires that the seller specify by its ID the referring item).

Item-to-Store-category. This type of rule allows the seller to specify for a particular referring item a category in that seller's eBay Store from which the cross-promotion items are drawn. The seller need not (and cannot) specify particular item IDs for the cross-promotion items. Up to four items are drawn from the specified custom Store category and used as the cross-promotion items (assuming that there are four items available in the category). Any type of listing (competitive-bidding or fixed-price) can be drawn from the designated category. Similarly, the referring item may be of any listing type.

Items that fail the basic query filters will not be used as cross-promotion items. If the specified category has more than four qualifying items, the determination of which four are used is based on sorts and filters in the seller's profile (where there are two sets of preferences, one for upsell items and one for cross-sell items). The items used for cross-promotion can be listed on the eBay site for any country, as long as the items are listed in the specified Store category. The referring item and the promoted Store category may – but do not have to be – on the same eBay site.

Store-category-to-Store-category. A Store-category-to-Store-category rule is applicable to any and all of the items listed in the designated triggering Store category. A seller need not do anything special for the rule to be associated with a newly listed item. Simply listing the item in the triggering category causes the rule to be associated with the item. A Store-category-to-Store-category rule continues to exist irrespective of the ending of any individual listings.

It is possible that when a referring item is displayed, multiple rule schemes could be applicable. In these cases, an order of precedence is applied to determine which rule scheme is activated and thus which cross-promotion items are displayed. For the upsell cross-promotion method, the order of precedence is:

- 1 Item-to-item
- 2 Item-to-store-category

This means that as many of the cross-promotion positions as possible are filled using items delivered by the item-to-item rule. Should there be fewer than four items returned by the item-to-item rule, then the remainder of the positions are filled by the applicable item-to-Store-category rule (if one has been defined). If that rule delivers too few items to fill all of the cross-promotion widget positions, then the remaining positions simply do not appear in the widget. (And if the total number of available items is fewer than two, the widget does not appear at all.)

For the cross-sell promotion method, the order of precedence is:

- 1 Item-to-item
- 2 Item-to-Store-category
- 3 eBay category algorithm

Cross-promotion widget positions are filled with items as described above for the upsell method. But for the cross-sell method, there is one additional criteria by which items are delivered for display in the cross-promotion widget. If the applicable item-to-item rule does not produce enough items to fill the widget (four items) and there is no item-to-store-category rule defined, then an algorithm is applied by eBay to select items for the remaining positions. These items come from a different category, based on a calculated buyer interest relationship between the referring item and the alternate category. Items returned by this eBay category algorithm are subject to the applicable query filters.

Cross-Promotion Placements

Merchandising placements are locations on (and off) the eBay site on which cross-promotions can appear. The list of placements shown below are those locations on which cross-promotions defined in an eBay API application can appear:

- **Active ViewItem Page** - top of page
- **ViewItem Page** - top of page
- **Closed ViewItem Page** - top of page
- **Closed ViewItem Page** - top of page
- **ViewItem Page** - description area

Assuming that an item's seller has opted to participate in Merchandising Manager (a preference setting on the seller's My eBay page), a user visiting one of the above cross-promotion placement locations is presented with other items the seller currently has listed.

Note that items are displayed on these placement locations (or not) within the constraints of the promotion method specified for the rule scheme that delivers the cross-promotion items. Basically, promotion methods are only applicable to certain placement locations and if the promotion method of a rule is not applicable to a placement location, then that rule is not applicable and is not used to deliver cross-promotion items to that placement location. (For more information on these constraints, see "[Cross-Promotion Methods](#)" on page 545.)

When the Active ViewItem Page (bid confirmation) is the means by which the viewer accesses the referring item, then the cross-promotion items appear at the top of the page, just under the user-specific status section. This page is displayed to a user immediately after the user has placed a bid on a competitive bid item listing (an auction). There are three variations of this page: bid confirmation – outbid, bid confirmation – reserve price not met, and bid confirmation – successful high bid (reserve

price met if there is a reserve price). Cross-promotion items only appear in this context if all of the following conditions are satisfied:

- The seller has opted to participate in Merchandising Manager.
- The bid is a successful high bid (no reserve price or reserve price is met).
- The item is not listed in the Mature Audiences category.
- The item is listed in a custom Store category (that is, the Store category ID is not 0).

When the ViewItem Page (where the buyer sees the Buy It Now confirmation) is the means by which the viewer accesses the referring item, then the cross-promotion items appear at the top of the page. This page is displayed when a buyer exercises the active Buy It Now option for the referring item. Cross-promotion items only appear in this context if all of the following conditions are satisfied:

- The seller has opted to participate in Merchandising Manager.
- The user has purchased the item through its Buy It Now option.
- The item is not listed in the Mature Audiences category.
- The item is listed in a custom Store category (that is, the Store category ID is not 0).

When the Closed ViewItem Page (winning bidder or buyer) is the means by which the viewer accesses the referring item, then the cross-promotion items appear at the top of the page. This page is displayed to a winning bidder when a competitive bid item listing (an auction) has ended. It is also displayed when a buyer has purchased a fixed-price item (listing types 7 and 9). Cross-promotion items only appear in this context if all of the following conditions are satisfied:

- The seller has opted to participate in Merchandising Manager.
- The listing has ended.
- The listing ended less than seven days prior.
- The user is the winning bidder.
- The item was purchased by the buyer (fixed-price listings).
- The item is not listed in the Mature Audiences category.
- The item is listed in a custom Store category (that is, the Store category ID is not 0).

When the Closed ViewItem Page (non-winning bidder, non-participant, or unknown user) is the means by which the viewer accesses the referring item, then the cross-promotion items appear at the top of the page. This is the ViewItem page as displayed for a closed listing to: one of the non-winning bidders from the listing, a user who did not participate in the listing (as a buyer or the seller), or whose relationship is undetermined (such as a non-participant user who is not logged in when the

page is viewed). Cross-promotion items only appear in this context if all of the following conditions are satisfied:

- The seller has opted to participate in Merchandising Manager.
- The listing has ended and within the past seven days.
- The item was purchased by the buyer (fixed-price listings).
- The user has purchased the item through its Buy It Now option.
- The item is not listed in the Mature Audiences category.
- The item is listed in a custom Store category (i.e., the Store category ID is not 0).
- The seller has an eBay Store subscription.
- One of the following is true:
 - The user viewing the referring item is unidentified.
 - The viewing user is identified and is one of the following:
 - A non-winning bidder from the referring item's listing
 - A user who did not participate in the referring item's listing

When the ViewItem Page (description area) is the means by which the viewer accesses the referring item, then the cross-promotion items appear immediately below the seller's description. Cross-promotion items only appear in this context if all of the following conditions are satisfied:

- The seller has opted to participate in Merchandising Manager.
- The seller has an eBay Store.
- The item is not listed in the Mature Audiences category.
- The item is listed in a custom Store category (that is, the Store category ID is not 0).
- Either of the following is true:
 - The referring item's listing is still active.
 - The referring item's listing has ended, but less than seven days prior.

Cross-Promotion Presentation Styles

The presentation style for a set of item cross-promotions governs how the items appear on the cross-promotion placement location. A presentation style is not the exact appearance of cross-promotion item placement. Rather, it is logic that governs how the cross-promoted items appear. For instance, the horizontal 4-item display might display fewer than four items if there are not four items in the designated Store category or if fewer than four items are returned because of the query filters. It is also possible that no cross-promotion items are available for the current context, in which case the cross-promotion widget does not appear on the cross-promotion placement at all.

Please note that the item needs to be listed with a Gallery image for the image to show up in the widget.

Assuming that there are items available for cross-promotion, these items are displayed on the cross-promotion placement being viewed by a user in a cross-promotion widget. This widget displays up to four items in a horizontal set of four merchandising slots, or positions. If an item-to-item rule scheme was used to retrieve the cross-promotion items, then the order in which the items are placed in the positions is influenced by the ordering specified by the seller when the rule was created. If it was a item-to-Store-category or Store-category-to-Store-category that produced the list of cross-promotion items, then the ordering of the items in the widget positions is determined by the filtering and sorting criteria the seller specified on his or her My eBay page.

The illustration below shows a horizontal 4-item display, as it would appear on the eBay site, fully populated with four cross-promotion items.



Note that if there are no cross-promotion items to display or the seller has not opted to participate in Merchandising Manager, this widget does not appear at all. Nor will the widget appear if there is only one item available. There must be two or more items (up to four) available for the cross-promotion widget to appear on the placement location.

Also, the item needs to be listed with a Gallery image for the image to show up in the widget.

When the widget displays cross-promotion items, it does so with a thumbnail image of the item, a link to the item's ViewItem page (using the item's title), the item's current price, and a button the user could click to bid on or purchase the item from the widget. These are all values that the `GetCrossPromotions` call returns.

The button that appears for an item (that a buyer uses to bid on or purchase the item) varies depending on what type of listing the item is. The table below shows the types of buttons that appear and the corresponding listing types. The text for the buttons is shown here in English. In actual usage, this text is localized for the language of the viewing user's site.

Table 33-2

Listing	Listing Types	Button
Auction	1, 2	Bid Now
Auction with Buy It Now still available	1	Buy It Now

Table 33-2

Listing	Listing Types	Button
Fixed-price (general)	9	Buy It Now
Fixed-price (Stores Inventory item)	7	Buy It Now
Real estate ad-format	6	<no button>

The thumbnail image that is shown for a cross-promotion item is the Gallery image specified when the seller originally listed the item. If no Gallery was specified, then the image used is the first URL specified as a Picture URL. If no images at all are available, then this space is filled by a spacer image. Like the item's title, the thumbnail image is a hypertext link to the item's **ViewItem** page.

The current price of the item is shown in the currency for the site on which the item was listed. While the display of the current price is the same no matter which listing type the cross-promotion, the meaning does vary. If the item is a competitive-bidding listing (listing types 1 and 2), then current price means the minimum allowable bid. If the item is a fixed-price listing (listing types 7 and 9), then current price means the purchase price. If the item is a Chinese listing with a still-available Buy It Now option, the current price refers to the price the buyer would need to pay to exercise that Buy It Now option.

Customizing the Display

You can also set a flag in your store preferences that determines how the widget is displayed. The flag is a value of **MerchDisplayCodeType**, and it displays the widget either with a default look and feel or the store theme colors.

The value of **MerchDisplayCodeType** is set from the **StoreType** object.

Defining Cross-Promotion Rules

Cross-promotion rules can be set up on the eBay site (through the Merchandising Manager interface). However, when a particular type of rule scheme needs to be defined varies depending on the rule type.

If the seller is setting up an item-to-item or an item-to-Store-category rule, then this needs to be done after the referring item has been successfully listed. This is because these rule types require that the item ID for the referring item be specified when the rule is defined.

The seller can set up a Store-category-to-Store-category rule scheme at any time, irrespective of item listing. For a Store-category-to-Store-category rule to be activated, the referring item merely needs to be in the specified referring Store category.

When defining a rule, the seller needs to specify:

- The item ID for the referring item (for item-to-item and item-to-Store-category rules only)
- The referring Store category ID (for Store-category-to-Store-category rules only)
- The promoted Store category ID (for item-to-Store-category and Store-category-to-Store-category rules only)
- Item IDs for one to four cross-promotion items (for item-to-item rules only)
- The promotion method (for all rule types)
- The promotion rule type (for all rule types)

Retrieving Cross-Promotion Rules

A seller can view a list of current rule schemes on the eBay site through the Merchandising Manager interface. Rules schemes can also be retrieved using the eBay API by calling the [GetPromotionRules](#) function.

Retrieving Cross-Promoted Items for Display with a Referring Item

When a referring item is viewed on the eBay site, the display of associated cross-promotion items occurs automatically (internal functionality of the eBay site). But in an eBay API application, this data must be retrieved before cross-promotions can be displayed.

Merchandising Manager cross-promotion item data is retrieved along with the data for the referring item when the [GetItem](#) function is used.

Cross-promotion item data can also be retrieved separately from the referring item's data by calling [GetCrossPromotions](#). The [GetCrossPromotions](#) function takes as one of its input arguments the item ID for the referring item. It is this value that determines which cross-promotion items (if any) are returned to be displayed with the item currently being shown to the user.

The following data must be displayed for each cross-promotion item. (See “[Cross-Promotion Presentation Styles](#)” on page 551 for specific display information for each value.)

- Image thumbnail
- Item title
- Current price
- Format button

Related Topics

- “[GetPromotionRules](#)” on page 855
- “[GetCrossPromotions](#)” on page 826
- “[GetItem](#)” on page 833
- “ [GetUser](#)” on page 876

Part IX

Specialty and International Sites

Working with Special eBay Sites	557
eBay Motors	559
Half.com	574
International Differences Overview	582
Working with Internationalization Issues	584
Working with Business Features and VAT	586
Field Differences for eBay Sites	590

Chapter 34

Working with Special eBay Sites

eBay supports a number of special eBay sites: eBay Stores, eBay Real Estate (US only), eBay Motors (US only), and Half.com.

An eBay Store is an online selling platform tied to eBay. A seller can build their own eBay Store, create custom categories to merchandise their Store, cross-sell and up-sell all of their items, use their own logo and HTML, and provide information about themselves, their Store and their policies.

Due to the wide variety of laws governing the sale of real estate, eBay Real Estate auction-style advertisements of real property do not involve legally binding offers to buy and sell. Instead, eBay Real Estate's auctions are simply a way for sellers to advertise their real estate and meet potential buyers. eBay Real Estate also offers sellers the option of advertising their property in an *Ad* format rather than an auction format. The real estate advertisement format was created for people selling real estate who want to get in contact with as many interested buyers as possible. As with Real Estate auctions, when the seller specifies a start price, it only serves as an asking price. No bidding takes place. With the Ad format, interested buyers fill out a contact form and their information is sent to the seller. Advertisement format listings do not result in feedback, because the format aims to generate multiple leads rather than a single high bidder.

The US eBay Motors site is a special automotive site run by eBay. It's designed to make selling and buying cars, trucks, SUVs, vans, motorcycles, boats, vehicle parts, and related items easy.

The Half.com site enables you to sell new and used Books, CDs, Movies, Games, and Game Systems. The items you list for sale will be seen by millions of potential buyers. You can list Half.com items for longer durations than eBay auction and fixed-price listings. Half.com takes a small commission when your item sells (you do not pay fees for unsold items).

The content of the sections below has been moved to new chapters to make them easier to find when you browse the documentation.

Working with Features for eBay Stores Sellers

This section has been moved to “[Managing eBay Stores](#)” on page 534.

Working with eBay Real Estate Listings

This section has been moved to “[eBay Real Estate Listings](#)” on page 367.

Working with US and CA eBay Motors Listings

This section has been moved to “[eBay Motors](#)” on page 559.

Working with Half.com Listings

This section has been moved to “[Half.com](#)” on page 574.

Chapter 35

eBay Motors

The eBay Motors US site is a specialty eBay site on which vehicles and related items can be offered for sale. In the context of the eBay Motors US site, vehicles include cars, trucks, motorcycles, fire engines, tractors, boats, and even helicopters. In addition to vehicles, sellers can list vehicle parts and vehicle-centric memorabilia on the eBay Motors US site.

The eBay Motors CA site follows many of the same rules and supports many of the same features as the eBay Motors US site.

Only the US maintains a separate site for vehicle listings, with its own site ID. That is, there are no special eBay vehicle sites for other countries. Instead, other eBay sites treat vehicles like any other category.

The majority of this chapter covers information that is specific to eBay Motors US and CA, unless otherwise specified.

This section covers these topics:

- “[Listing US and CA eBay Motors Items](#)” on page 559
- “[Searching for eBay Motors US Listings](#)” on page 571
- “[Working with eBay Motors Germany Listings](#)” on page 572
- “[Working with eBay Motors Australia Listings](#)” on page 573

Listing US and CA eBay Motors Items

Certain rules for listing items to or retrieving items from the eBay Motors US site are different from the rules for listing to the main eBay US site.

If you have downloaded this documentation locally, please refer to the documentation on the eBay Developers Program Web site for the latest rules and details.

eBay Developer Documentation Center
<https://www.x.com/developers/ebay/documentation-tools/>

To list a vehicle on the US eBay Motors site, make sure you specify a site ID of 100 (not 0). Then specify the following fields:

- The correct category ID for your vehicle. This will also pre-populate the vehicle make and model. See “[Specifying Categories for eBay Motors Listings](#)” on page 560 and “[Specifying the Vehicle Manufacturer and Model](#)” on page 567.
- The listing format, such as fixed-price, auction, or classified. See these sections for details:
 - “[Specifying a Listing Format for eBay Motors Listings](#)” on page 561
 - “[Local Market Listings](#)” on page 562
- The title of the listing, which appears as a listing subtitle to buyers. See “[Specifying a Subtitle for a Motors Vehicle Listing](#)” on page 568.
- The Vehicle Identification Number (VIN). See “[Specifying the Vehicle Identification Number \(VIN\)](#)” on page 564.
- The payment methods, including your requirements for a vehicle deposit. See these sections:
 - “[Payment Methods for Motors](#)” on page 568
 - “[Specifying a Vehicle Deposit and Deposit Amount](#)” on page 568
 - “[Requiring Immediate Payment of a Vehicle Deposit](#)” on page 569
- Common details about the vehicle that buyer typically use to distinguish models, such as the engine type. See “[Using Item Specifics in eBay Motors US Listings](#)” on page 565.
- Information about how you will ship or deliver the vehicle. See “[Shipping Details for eBay Motors](#)” on page 562.
- If you have an eBay store, your eBay Motors listings will appear in your eBay store. See “[Managing eBay Stores](#)” on page 534.

In addition, the following sections may be of interest:

- eBay offers vehicle purchase protection for all vehicles. Sellers do not need to specify this. See “[Vehicle Purchase Protection](#)” on page 570.
- Revising vehicle listings is similar to revising other listings. See “[Revising US Motors Listings](#)” on page 570.
- If you are listing vehicles on eBay Germany, see “[Cross-Promoting Vehicles on the Mobile.de Site](#)” on page 572 for additional options.

Specifying Categories for eBay Motors Listings

eBay Motors has a distinct category hierarchy in which items are listed, and categories and their IDs are not the same as on any other eBay site. When you retrieve the eBay Motors category hierarchy and related category data using [GetCategories](#), [GetCategoryFeatures](#), and [GetCategorySpecifics](#), you need to specify the eBay Motors site instead of the main eBay site. To do this, specify a value of 100 in the X-EBAY-API-SITEID header (for the XML API) or in the request URL (for the SOAP API) (see “[Routing the Request \(Gateway URLs\)](#)” on page 87) and set [CategorySiteID](#) to “100”.

Similarly, when you list an eBay Motors US item using a call like [AddItem](#), you need to specify the eBay Motors site instead of the main eBay site. To do this, pass a site ID of 100 as described above, and set [Item.Site](#) to [eBayMotors](#).

Listing eBay Motors Items in Two Categories

The rules for listing in one or two categories depends on the category:

- In the US, vehicles can only be listed on eBay Motors and only in one category (not in two categories). You can't list eBay Motors US vehicles on international sites or the main US site. If you want to list cars, motorcycles, boats, or other vehicles on an international eBay site, list them in the regular automotive or vehicle categories for those sites. For a list of categories that support vehicles on an international site, use [GetCategories](#) or look at the list of categories posted on the appropriate Web site. Use [GetFeatures](#) to determine whether an international site's automotive categories also support custom Item Specifics. Use [GetCategorySpecifics](#) to retrieve recommendations for custom Item Specifics.
- On the eBay Motors US site, Parts & Accessories can be listed in two categories.

Furthermore, to make it easier for buyers to find Parts & Accessories, you can combine a Motors Parts & Accessories category and an eligible eBay category as the primary or secondary category, as long as the primary category is associated with the listing site. (Most eBay categories except Real Estate, Mature Audience, and Business & Industrial categories are eligible to be combined with Motors Parts & Accessories categories.)

For example, if you are listing a Harley Davidson shirt on the main eBay US site (site ID 0), you can specify an eBay clothing category as the primary category, and specify a Motors Parts & Accessories category as the secondary category. We'll look for the secondary category ID on eBay first. If we don't find it there, we'll look for the ID on the Motors site. (Leaf category IDs are unique across the US and eBay Motors US sites.)

Conversely, if you are listing the shirt on the eBay Motors US site (site ID 100), you can specify a Parts & Accessories category as the primary category, and specify an eBay clothing category as the secondary category. In this case, we'll look for the secondary category ID on the Motors site first. If we don't find it there, we'll look for the ID on the main eBay site.

Specifying a Listing Format for eBay Motors Listings

eBay Motors items within the Parts & Accessories and the Everything Else categories can be listed in Auction or Basic Fixed-Price formats.

When you list nationally, eBay Motors items within vehicle categories can be listed in an auction-style formats, but cannot be listed in the Fixed Price format.

See "[Local Market Listings](#)" on page 562 for listing format rules when you list vehicles as Local Market listings.

eBay Motors and eBay Stores

Items that an eBay Stores seller lists to the eBay Motors US site also appear in the seller's eBay Store. As with any listing, Stores sellers can define custom store categories for their eBay Motors listings.

See “[Managing eBay Stores](#)” on page 534.

Shipping Details for eBay Motors

For eBay Motors listings, shipping details depend on the category:

- For Parts & Accessories and the Everything Else categories, all of the normal shipping option codes are available. If you use [GeteBayDetails](#) to retrieve shipping details for Parts & Accessories for the US, use site ID 0 (not site ID 100).
- For vehicles, use [Item.ShippingTermsInDescription](#), and explain the seller’s shipping arrangements and return policy in the [Item.Description](#) field and the [Item.ShippingDetails.PaymentInstructions](#) field.
- To indicate that the buyer is responsible for obtaining and transporting the car, motorcycle, or part, use [Item.BuyerResponsibleForShipping](#). In this case, if you list a vehicle on the eBay Motors US site and you want your listing to be visible on the eBay Canada site, set [Item.ShipToLocations](#) to [CA](#). Similarly, If you list a vehicle on the eBay Canada site and you want your listing to be visible on the eBay Motors US site, set [Item.ShipToLocations](#) to [US](#). Otherwise, do not specify [Item.ShipToLocations](#) when the buyer is responsible for shipping.

Local Market Listings

Sellers can list vehicles for the eBay Motors Local Market only. The Local Market is defined as the area within a set radius about a specified postal code. Local Market listings can be Best Offer Only, if the Listing Type is set to ‘Lead Generation’. Buyers can retrieve Local Market vehicle listings in searches only when they supply a postal code.

Note: Local Market listings are supported for vehicle categories on the eBay Motors US site only and for US postal codes only.

Options to list as ‘Fixed Price’, ‘Buy It Now’ or ‘Buy It Now + Best Offer’ are no longer available for Local Market listings. If you have existing listings that use these categories, they will remain on the site and you can revise them until they expire.

Adding Local Market Listings

To add a Local Market listing, use [Item.ListingDetails.LocalListingDistance](#) to specify a distance (the radius about the item location that constitutes the Local Market) and set [Item.ListingType](#) to “FixedPriceItem” in an [AddItem](#) request. Buyers can set the local listing distance to one of the preset values, such as 25, 50, 75, or 100. Use [FeatureID.LocalListingDistances](#) in [GetCategoryFeatures](#) to retrieve a list of valid values for local listing distance by category.

You can specify additional seller information such as: Address/Dealer Location, Telephone number (CallBright), and Contact Hours using an [AddItem](#) request. If the seller does not specify an address/dealer location, the vehicle location specified in the shipping section will be used.

Optionally, the listing can use the Lead Generation selling format by setting [Item.ListingType](#) to LeadGeneration and setting [Item.BestOfferDetails.BestOfferEnabled](#) to true. This creates a Best Offer Only listing that requires a [StartPrice](#), which will be displayed as the Advertised Price to buyers.

Note the following when creating a Local Market listing:

- Local listing distance must be specified with the added [Item.LocalListingDistance](#) field.
- Location of the vehicle is specified with the [Item.PostalCode](#) field.
- One picture is required for Local Market listings.
- Use of URL imprints or watermarks on pictures is prohibited.
- Use of video or flash must conform to eBay listing requirements.
- Email addresses or links are not permitted in the listing. You may provide links through the Ask Seller a Question form.
- Only Callbright telephone numbers are allowed in Local Market listings. Callbright numbers should NOT be used in National Market listings.
- [Item.ListingType](#) field must be set to "LeadGeneration" (for Best Offer Only listings).
- Use of external lead forms is prohibited. All lead information must be acquired using the eBay lead form.
- If you are creating a Best Offer Only listing:
 - Specify a start price with the [Item.StartPrice](#) field
 - Set the [Item.ListingType](#) field to "LeadGeneration"
 - Set [Item.BestOfferDetails.BestOfferEnabled](#) to true
- The following fields are invalid for local listings:
 - [Item.PrivateListing](#)
 - [Item.ListingEnhancement](#)
 - [Item.GiftIcon](#)
- If a local listing distance is specified and any of the values for these fields is not null, a warning is returned.
- Local listings support a listing duration of Days_30 and GTC. GTC is only allowed for sellers with Local Market subscriptions.

Note: The [GetCategoryFeatures](#) call does not return listing duration information for Local Market Listings.

Revising and Relisting Local Market Listings

Note the following call-specific differences when revising or relisting a Local Market listing:

- **ReviselItem** cannot be used to change a non-local listing to a local listing, and vice versa.
- **ReviselItem** can be used to change the value for **Item.LocalListingDistance** to one of the valid values supported by the given category.
- **RelistItem** can be used to relist a non-local listing as a local listing, by specifying a valid local listing distance with **Item.LocalListingDistance**.
- **RelistItem** can be used to relist a local listing as a non-local listing by using **DeletedField** to delete **Item.ListingDetails.LocalListingDistance**.

Relisting local vehicle listings is subject to the current restrictions for relisting items in eBay Motors, unless otherwise specified.

Best Offer Only

Best Offer Only listings are created using the Lead Generation format. This special format picks up all of the current functionality of a listing for which Best Offer has been enabled.

For example, **GetBestOffers** and **RespondToBestOffer** handle a Best Offer Only local market listing the same as the other listings that support best offers.

In the **GetCrossPromotions** call response, the **PromotionPriceType** field is set to "BestOfferOnlyPrice" when a promoted item is a Best Offer Only listing.

Additionally, **GetMyeBayBuying** handles a Best Offer Only local market listing the same as the other listings that support best offers. Local market listing items will be returned in **BestOfferList** of the **GetMyeBayBuying** response, and included in the **BuyingSummary.BestOfferCount** value. Likewise, **GetMyeBaySelling** handles a Best Offer Only local market listing the same as the other listings that support best offers and returns the listing in relevant lists: **ScheduledList**, **ActiveList**, **SoldList**, or **UnsoldList**.

Sellers cannot end a Best Offer Only local vehicle listing early if there are active best offers.

Retrieving Lead Information

When a buyer is interested in a Classified Ad vehicle listing in eBay Motors or Motors categories, he can leave information for the seller via the Ask Seller a Question or eBay lead form on the View Item page. This *lead* information includes data like name, email address, telephone, best time to contact, trade-in information, interest in financing, and the buyer's time frame for purchasing.

The seller can use the **GetAdFormatLeads** request to retrieve lead data for his listings.

Specifying the Vehicle Identification Number (VIN)

For the US eBay Motors site, a Vehichle Identification Number (VIN) is required for vehicles from model year 1981 or later. This VIN appears as VIN Number in the Item Specifics section of the View Item page.

Specify this data in **Item.VIN** (or **Item.VRM** for the UK site) instead of in item specifics.

For rules that are specific to Australia eBay Motors, see “[Specifying the VIN and Registration Date for eBay Australia](#)” on page 573.

eBay uses the VIN (or VRM) to create a link to the vehicle’s Vehicle History Report.

In [GetItem](#), the vehicle history report is returned in [Item.VINLink](#) (or [Item.VRMLink](#) for the UK site). In the Shopping API, [GetSingleItem](#) and [GetMultipleItems](#) return the report in [VhrURL](#). If no vehicle history report is available, eBay may replace the Vehicle History Report link on the eBay Web site with a plain-text copy of the VIN.

Using Item Specifics in eBay Motors US Listings

On the eBay Motors US site, you use a combination of vehicle-specific fields and custom Item Specifics to fill in vehicle details like Mileage and Exterior Color.

The eBay Motors US Web site UI supports two listing flows:

- You can list the standard way by filling in Item Specifics yourself. The API supports this flow.
- (*Not supported in the API*) On the Web site, you can list by using a VIN to include product details. We refer to this as the “VIN catalog” flow. The API does not support this flow (although you are required to pass inVIN in an [AddItem](#) request, for modern vehicles.)

If you are using the XML API or the SOAP API, the next sections describe the basic process you use to list vehicles to eBay Motors US.

Using GetCategories to Retrieve eBay Motors Categories

First, use [GetCategories](#) to retrieve all the available eBay Motors categories. Be sure to specify the eBay Motors US site instead of the main eBay site. To do this, specify a value of 100 in the X-EBAY-API-SITEID header (for the XML API) or in the request URL (for the SOAP API). (See “[Routing the Request \(Gateway URLs\)](#)” on page 87.)

Example 35-1 shows a typical [GetCategories](#) request, using the XML API.

Example 35-1 GetCategories Request (XML API)

```
<?xml version="1.0" encoding="utf-8"?>
<GetCategoriesRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>YOURTOKENHERE</eBayAuthToken>
  </RequesterCredentials>
  <CategorySiteID>100</CategorySiteID>
  <DetailLevel>ReturnAll</DetailLevel>
  <ViewAllNodes>true</ViewAllNodes>
</GetCategoriesRequest>
```

As usual for this call, store the response locally. In most cases, eBay Motors categories are associated with specific makes and models, like “Pontiac : Trans Am”. You will use a category ID to identify the make and model in [Item.PrimaryCategory.CategoryID](#) when you list the item.

Example 35-2 shows the Pontiac category and the Trans Am category in the [GetCategories](#) response for the eBay Motors US site.

- Category 6377 (Pontiac) specifies the make.
- Category 6427 (Trans Am) specifies the model, and it also specifies that its parent ID is 6377 (Pontiac), so we know the make.

[GetCategories](#) doesn't specifically indicate "make" or "model", but for vehicles the leaf category is the model. (The example also shows that 6377's parent is 6001, which is Cars & Trucks, so we know this is a vehicle category.).

Example 35-2 Categories in the GetCategories Response (XML API)

```
<Category>
  <BestOfferEnabled>true</BestOfferEnabled>
  <AutoPayEnabled>true</AutoPayEnabled>
  <CategoryID>6377</CategoryID>
  <CategoryLevel>3</CategoryLevel>
  <CategoryName>Pontiac</CategoryName>
  <CategoryParentID>6001</CategoryParentID>
  <Expired>false</Expired>
  <IntlAutosFixedCat>false</IntlAutosFixedCat>
  <LeafCategory>false</LeafCategory>
  <Virtual>false</Virtual>
  <SellerGuaranteeEligible>true</SellerGuaranteeEligible>
  <ORPA>false</ORPA>
  <LSD>true</LSD>
</Category>

<Category>
  <BestOfferEnabled>true</BestOfferEnabled>
  <AutoPayEnabled>true</AutoPayEnabled>
  <CategoryID>6427</CategoryID>
  <CategoryLevel>4</CategoryLevel>
  <CategoryName>Trans Am</CategoryName>
  <CategoryParentID>6377</CategoryParentID>
  <Expired>false</Expired>
  <IntlAutosFixedCat>false</IntlAutosFixedCat>
  <LeafCategory>true</LeafCategory>
  <Virtual>false</Virtual>
  <SellerGuaranteeEligible>true</SellerGuaranteeEligible>
  <ORPA>false</ORPA>
  <LSD>true</LSD>
</Category>
```

Using [GetCategoryFeatures](#) and [GetCategorySpecifics](#) to Retrieve Item Specifics

Details about motorcycles are different from details about Cars & Trucks . A motorcycle would have details like "Engine Size (cc)", and truck would have details like "Cab type". In the API, these details are called Item Specifics.

To determine whether a vehicle category supports custom Item Specifics, call [GetCategoryFeatures](#) and look for [ItemSpecificsEnabled](#) in the response, for the specified category. If the category supports custom Item Specifics, call [GetCategorySpecifics](#) to retrieve the recommendations and any related rules. See "[Working with Custom Item Specifics](#)" on page 481.

Specifying the Vehicle Manufacturer and Model

When you submit a Cars & Trucks listing to the eBay Motors US site, we strongly recommend that you avoid specifying the Manufacturer and Model item specifics in most cases. In most cases, this information will be pre-filled based on the category you select.

- If you specify a primary or secondary category that represents a distinct manufacturer and model, eBay automatically populates the correct Manufacturer and Model item specifics. For example, if the category is 6015 (Porsche > Boxster), the Manufacturer is automatically set to "Porsche" and the Model to "Boxster."

Therefore, do not specify the Manufacturer and Model item specifics when you specify a category that is associated with a distinct manufacturer and model. If you do, you may receive an error if these item specifics do not correctly match the category. For example, if the category ID is 6015, then the Manufacturer must be "Porsche" and the Model must be "Boxster". In this case, if you specify that the Model is "912", you will receive an error.

- If your vehicle is not represented by a distinct eBay Motors category, you may specify a generic ID (e.g., 6058 for Porsche > Other Models) in the appropriate category property and specify the Model item specific. The value of your Model item specific will be appended to the title of your listing.

Use caution when including the Manufacturer and Model item specifics with generic categories. If you specify a generic ID in the category property (e.g., 6058 for Porsche > Other Models) but you include Manufacturer and/or Model item specifics that match a category with a predefined distinct model, eBay will list the vehicle in a more generic category (e.g., All Models) rather than the distinct category implied by the item specifics you specified. In this case, buyers might not find the listing where the seller intended to place it.

- You can use the Submodel item specific to append more details about the model (up to 12 characters). eBay adds a space and appends the text to the "Manufacturer : Model" value in the listing title at the top of the View Item page, in the Item Specifics section of the page, and in search results. (eBay does not actually store the submodel text as part of the title, so [Item.Title](#) does not include the text in [GetItem](#).)
- If you are listing in a Motorcycles category, eBay handles the Manufacturer and Model item specifics for Motorcycle categories so that they are similar to the Cars & Trucks category format. For example, instead of returning Manufacturer="BMW : F-Series" and Model="F-Series", [GetItem](#) will return Manufacturer="BMW" and Model="F-Series".

As with Cars & Trucks, you should not specify Manufacturer and Model item specifics when a motorcycle is represented by a distinct eBay Motors category (i.e., a category other than "All Models" or "Other"). eBay pre-fills these item specifics based on the parent and leaf category.

If the seller specifies a Model item specific for a motorcycle in a distinct category, eBay may replace the value and pre-fill the Model item specific based on the leaf category (e.g., F-series). If this occurs, eBay may move the Model value the seller specified into the Submodel item specific so that no data will be lost.

This means the item specific that are returned in [GetItem](#) may be different from the corresponding item specific you specify in [AddItem](#) and related calls.

Specifying a Subtitle for a Motors Vehicle Listing

On the eBay Motors site, the listing title is filled in automatically by eBay for certain vehicle categories (e.g., Cars & Trucks).

Use the [Item.SellerProvidedTitle](#) field to specify a seller's personalized title that will appear on the View Item page and that can be used to differentiate the item in search results. Choose keywords that help distinguish your vehicle in search results, such as color, condition, body style, and unique options.

For Cars & Trucks, the [SellerProvidedTitle](#) is a required field.

The subtitle appears in the Item Specifics section of a listing.

Payment Methods for Motors

Payment methods for items are defined by using [Item.PaymentMethods](#), as you do with other listings. This code list includes several methods that can be used for items on the eBay Motors US site. There are specialized payment provisions and methods for selling vehicles on the eBay Motors US site (except in the Parts and Accessories category) and on the eBay Canada site for motors.

When you list a vehicle, at least one full payment method that applies to vehicle listings is required. Specifically, one or more of the following payment methods must be used: [MOCC](#), [PersonalCheck](#), [LoanCheck](#), [CashInPerson](#), and/or [PaymentSeeDescription](#). The [LoanCheck](#) and [CashInPerson](#) payment methods are specific to vehicle listings. Buyers cannot make full payments by credit card or PayPal.

Note: If a vehicle listing offers PayPal or a credit card, then the listing must also specify deposit (because PayPal and credit cards cannot be used for full payments). See “[Specifying a Vehicle Deposit and Deposit Amount](#)” on page 568.

[DaysToFullPayment](#) can be used to specify the number of days after the auction closes that the buyer can take to pay for the item. The choices are 3, 7, 10, and 14, and the default is 7.

On the eBay Motors US site, a deposit can be specified. (The eBay Motors Canada site doesn't support deposits). If you specify a deposit, you can also use [HoursToDeposit](#) to specify the number of hours after an auction closes that the buyer can put down a deposit on the item. See “[Specifying a Vehicle Deposit and Deposit Amount](#)” on page 568.

Specifying a Vehicle Deposit and Deposit Amount

When you require a vehicle deposit, the completed listing displays an option (such as a **Pay** button) to enable the buyer to pay the deposit online.

Use the [Item.PaymentDetails.DepositAmount](#) field to specify the amount of a vehicle deposit (a non-zero value up to \$2000.00) . Vehicle deposits must be paid for through PayPal.

If requiring a deposit on a vehicle, the seller must:

- Use [HoursToDeposit](#) to specify the number of hours after an auction closes that the buyer can pay the deposit on the vehicle. The choices are 24, 48, and 72, and the default is 48.
- Use [DaysToFullPayment](#) to specify the number of days after an auction closes that the buyer has to pay off the remaining balance on the vehicle. The choices are 3, 7, 10, and 14, and the default is 7.
- Specify [PayPal](#) (in [Item.PaymentMethods](#)) as one of the payment methods. It will be used for the deposit. (PayPal enables buyers to pay via credit card, but you cannot directly specify particular credit cards like [AmEx](#) for the deposit.)

Note: Unlike other listings, PayPal is not automatically added to Motors vehicle listings even if the seller has a PayPal preference set.

Also, the seller must have a linked PayPal account.

- Pass the seller's registered PayPal email address in [PayPalEmailAddress](#) (as usual).
- Include one or more of the following payment methods for the balance:
 - [MOCC](#)
 - [PersonalCheck](#)
 - [LoanCheck](#)
 - [CashInPerson](#)
 - [PaymentSeeDescription](#)

Also see "[Requiring Immediate Payment of a Vehicle Deposit](#)" on page 569.

When revising a vehicle listing, additional validation rules apply:

- If the listing has no bids:
 - The seller can add a deposit or lower an existing deposit amount.
 - If the seller requires Immediate Payment, the seller cannot increase the deposit amount.
 - If Immediate Payment is not required, the seller can increase the deposit.
 - The seller can remove the Immediate Payment option, if any.
- If the listing has bids:
 - The seller can lower the deposit amount (if any).
 - The seller cannot increase the deposit amount or add a deposit.
 - The seller cannot remove the Immediate Payment option (if any).

Requiring Immediate Payment of a Vehicle Deposit

When listing a vehicle on the eBay Motors US site, a seller who is eligible to offer Immediate Payment can choose to require Immediate Payment of the vehicle deposit for eligible listings.

If the seller chooses to require Immediate Payment of the deposit, follow all the same rules that are described in “[Specifying a Vehicle Deposit and Deposit Amount](#)” on page 568, and also set **AutoPay** to true (to indicate that Immediate Payment is required).

Vehicle Purchase Protection

eBay offers Vehicle Purchase Protection, which ensures that you receive the vehicle you paid for, with coverage up to \$20,000 against fraud or material misrepresentation. (The seller does not choose this. It is offered by eBay.)

For information about vehicle categories that support Vehicle Purchase Protection, please refer to the following help pages:

Vehicle Purchase Protection (US)

<http://pages.motors.ebay.com/services/purchase-protection.html>

Passenger Vehicle Purchase Protection (CA)

<http://pages.ebay.ca/ebaymotors/help/education/buying/purchase-protection.html>

On the eBay Motors US site, when you use **GetItem**, the Purchase Protection item specific is returned also if no Vehicle Purchase Protection is available, or true if it is available. Other sites that support this feature and that use Item Specifics for vehicles return similar information.

Listing eBay Motors US Items in eBay Stores

Items that an eBay Stores seller lists to the eBay Motors US site (site ID of **eBayMotors**) also appear in the seller’s eBay Store.

eBay Motors items within the Parts & Accessories and the Everything Else categories can be listed in Chinese Auction, Store Inventory, or Basic Fixed-Price formats. eBay Motors items within vehicle categories can be listed in Chinese Auction, or Fixed Price formats, but cannot be listed in the eBay Store Inventory format.

Revising US Motors Listings

See “[Revising US eBay Motors Listings](#)” on page 385.

Also see “[Revising and Relisting Local Market Listings](#)” on page 564.

Short-Term Service Agreement (Limited Warranty)

Important: This feature is not currently supported.

If you pass **LimitedWarrantyEligible** in your request, it is ignored as of June 2010. The following information is being provided as background information for developers who are removing support for this feature from their applications.

For certain vehicles, eBay previously offered a free 1 month/1,000 mile Short-Term Service Agreement (also referred to as a limited warranty). Only vehicles that are not disqualified as having

certain modifications and that are not classified as special vehicle types (e.g., Corvette ZR1) were eligible for this limited warranty.

When listing a Motors vehicle, your application was required to provide the seller with an opportunity to indicate whether the vehicle has not been modified and is not a special type:

- Your application needs to ask the seller whether the vehicle has been modified.
- Based on the item's category, the eBay code will check to determine if a vehicle fits the "Special Vehicle Type" rules, which would disqualify a vehicle. Your code does not need to provide this check. However, your application needs to ask the seller to indicate whether the vehicle is a special vehicle type.

For a complete list of vehicle modifications and special vehicle types that are not eligible for the limited warranty, see the eBay Motors site:

Short Term Service Agreement Coverage Plan Details

http://pages.motors.ebay.com/help/buyandsell/lw_details.html

You were required to present the following options to the seller as part of the listing process. If you present the options as radio buttons, use this text (where the parentheses "()" represent radio buttons):

Please check one of the following:

- (X) The vehicle listed has **no modifications** and is **not a special vehicle type**.
() The vehicle listed **has been modified** and/or **is a special vehicle type**.

In listing requests, the **LimitedWarrantyEligible** flag was used to indicate the seller's selection. If **LimitedWarrantyEligible** is passed in, it means your application has presented the seller with options to indicate this information. If the value is **true**, the seller has verified that the vehicle has not been modified and is not a special type. If **false**, the seller has verified that vehicle has been modified or it is a special type.

Passing a value of **true** did not guarantee that eBay will offer a limited warranty. It only made the listing eligible for consideration.

On the eBay Motors US site, when you use **GetItem**, the Limited Warranty attribute (26738) is returned with a value of **-14 (No)** if the limited warranty is not available (it previously returned **-13 (Yes)** if it was available). Other sites that support this feature and that use Item Specifics for vehicles return similar information. Use **GetAttributesCS** to determine the applicable attribute and value IDs.

Searching for eBay Motors US Listings

To search for eBay Motors listings, use **FindItemsAdvanced** in the Finding API.

Be sure to specify the eBay Motors site (ID 100).

FindItemsAdvanced in the Finding API

<http://developer.ebay.com/DevZone/finding/CallRef/findItemsAdvanced.html>

GetMultipleItems in the Shopping API

<http://developer.ebay.com/DevZone/Shopping/docs/CallRef/GetMultipleItems.html>

Use category IDs for the eBay Motors site to search against a vehicle's make and model. A category ID of 6001 can be used to search all vehicle categories, 6024 to search just motorcycle categories, 6028 to search all parts categories, and 6038 to search other categories. These generic categories can also be used if the site ID is 0 (main eBay US site), in which case they automatically direct the search to eBay Motors. However, if you use site ID 0 instead of site ID 100, the API might not return data that is specific to the eBay Motors site, such as the vehicle's mileage and year.

Working with eBay Motors Germany Listings

This section provides brief notes about working with listings on eBay Motors Germany.

Cross-Promoting Vehicles on the Mobile.de Site

This listing upgrade is applicable to listings on the eBay Germany site only.

Mobile.de (a German Motors classifieds platform) cross-promotes items from eBay Germany's Car, Motorcycle, and Special Vehicle categories by displaying them to users who are browsing and searching on the mobile.de site. An eBay seller can purchase a listing upgrade to allow mobile.de to include the listing in mobile.de search results. To be eligible, the item must be a vehicle listed on the eBay Germany site (site ID 77) in a subcategory of one of the following mobile.de search-enabled categories.

Table 35-1 Mobile.de Search-Enabled Categories

Category Name	Category ID
Automobile	9801
Spezielle Fahrzeuge	46159
Motorräder	9804

Usage of this feature is optional. It does not affect the visibility of the listing on eBay.com sites.

In `AddItem` and other requests that list items, pass the `MotorsGermanySearchable` flag with a value of `true` to indicate that the seller wants to purchase this listing upgrade. If the item and category are eligible, the item will be included in eBay search results on the mobile.de site. This flag is ignored if specified for other categories. It will trigger an error if specified for other sites.

This flag can be added to an active listing at any time by using `Reviseltem`, regardless of whether the listing has bids or how soon the listing ends, as long as the item and category are eligible for this feature. Once this flag set to true for a listing, you cannot reset it to false when revising an item that is still listed in a mobile.de search-enabled category.

You can set the **MotorsGermanySearchable** flag to false (and remove the listing from mobile.de site search results) when relisting the item, regardless of the category. Similarly, if the seller changes the category to one that is not mobile.de search-enabled, eBay resets the **MotorsGermanySearchable** flag to false.

Working with eBay Motors Australia Listings

This section provides brief notes about working with listings on eBay Motors Australia.

Specifying the VIN and Registration Date for eBay Australia

eBay Motors Australia uses different custom Item Specifics than the US, but you use the same calls and approach.

The VIN is a unique set of 17 characters found on all Australian vehicles manufactured after 1989.

When you specify **VIN**, always also specify the Date of 1st Registration in item specifics. There is a dependency between the VIN and the Date of 1st Registration. If you don't specify the Date of 1st Registration:

- The link to the Vehicle History Report can't be generated correctly.
- The VIN may not appear at all in the listing.

Chapter 36

Half.com

The Half.com site enables you to sell new and used Books, CDs, Movies, Games, and Game Systems. The items you list for sale will be seen by millions of potential buyers. You can list Half.com items for longer durations than eBay auction and fixed-price listings. Half.com takes a small commission when your item sells (you do not pay fees for unsold items).

Half.com by eBay
<http://www.half.ebay.com/>

Buyers can browse and search the Half.com site to find your Half.com listings.

The sections below assume that you are familiar with basic eBay concepts such as items, order line items, and orders.

The Sandbox does not support Half.com. It does not support registering as a Half.com seller, listing Half.com items, or processing Half.com orders or payments.

The Half.com Listing Workflow

When you use the Production environment (i.e., the live Half.com site), the process below summarizes the typical workflow for working with Half.com listings via the API. The sections that follow provide more information that you should be aware of when you are developing applications that work with Half.com items, orders, and payments.

- 1 Register as a Half.com seller. See “[How to Become a Half.com Seller](#)” on page 575. After registering, you can configure standard information that will apply to all your listings. (You only need to register once.)
- 2 Use [AddItem](#) to list an item to Half.com. In the request, be sure to specify a value of `Half` for the [ListingType](#). Otherwise, the item will be listed to eBay.com.
- 3 Once you list an item, Half.com provides the cover art, review, and product information from its extensive database.
- 4 Use [RevisedItem](#) or [EndItem](#) if you need to modify or delete any of your Half.com listings. Unlike other eBay.com items, you need to also specify a value of `Half` for the [ListingType](#). Otherwise the correct logic will not be used, and the RevisedItem could return strange errors, or you may see unwanted side effects on the item (even if no errors are returned).
- 5 A buyer uses the Half.com Web site to purchase one or more items. All items in their shopping cart are combined into a single order. The order can contain multiple items from multiple sellers.
- 6 When the buyer pays for the order, Half.com creates a pending payment for each line item in the order. Each payment includes the relevant seller’s share of the sale amount, a reimbursement for the seller’s shipping costs, and Half.com’s commission.

- 7 Periodically (e.g., once per day), use [GetOrders](#) to retrieve your current unshipped orders, and use [GetSellerPayments](#) to retrieve information about your pending payments. Ship items to buyers as appropriate, and use [CompleteSale](#) to mark the items as shipped. Use [LeaveFeedback](#) to leave feedback for the buyer.
- 8 Within approximately a week after the pay period ends, Half.com sends (transfers or deposits) all pending payments to your financial institution and marks the payments as paid.
- 9 Periodically (e.g., twice per month), use [GetSellerPayments](#) to retrieve information about your paid payments.
- 10 As necessary, use [IssueRefund](#) to issue refunds to buyers for returned items.

How to Become a Half.com Seller

An eBay member can use their eBay user ID when listing items to Half.com. However, that user ID must be also be registered as a Half.com seller. For information on becoming a Half.com seller (in the Production environment), see the Half.com online help:

How to Become a Half.com Seller (online help)
http://half.ebay.com/help/index.cfm?helpsection=createacct_sell

The API does not provide a programmatic way to determine whether a seller is authorized to list Half.com items.

The Sandbox does not support Half.com. It does not support registering as a Half.com seller, listing Half.com items, or processing Half.com orders or payments.

Half.com Payments and Commissions

Instead of paying a seller directly, buyers pay Half.com for each order, and then Half.com reimburses the seller through bulk payments on a periodic basis. Instead of an insertion fee, Half.com charges the seller a small commission.

Half.com Seller's Handbook: Getting Paid
<http://half.ebay.com/help/index.cfm?helpsection=getpaid>

Half.com Pay Periods and Payments

Each payment is for one line item in one order. An order can contain line items for multiple items from multiple sellers, but a seller can only retrieve their own payments.

Sales and Refunds

Payments from Half.com to the seller are *sale* payments. Payments from the seller to the buyer are *refund* payments.

The financial value of a sale payment is typically based on an amount that a buyer paid to Half.com for an order line item, with adjustments for shipping costs (a shipping reimbursement for the seller) and Half.com's commission.

For refund payments, Half.com allows a maximum of the original item sale price (order line item price plus original shipping reimbursement) plus return shipping costs (the amount the buyer paid to return the item). Typically, the return shipping cost is based on the current cost of shipping the individual item (not the discounted cost calculated during the original checkout for a multi-item order). You can also issue a partial refund for the amount you want the buyer to receive.

Pay Periods

For most sellers, each month contains two pay periods:

- One from the 1st to the 15th of the month
- One from the 16th to the last day of the month

Sellers can refer to their account information on the Half.com site to determine their pay periods. (You cannot retrieve a seller's pay periods by using the API.)

When a buyer makes a purchase and an order is created, Half.com creates a payment for the seller and marks it as Pending in the seller's Half.com account. Within a certain number of days after the pay period ends, Half.com settles payments for that period and marks each completed payment as Paid. See the Half.com Web site online help for more information about how payments are managed.

You can use [GetSellerPayments](#) to retrieve your pending and paid payments for sales and refunds. When you use this call, you should understand these notions:

- Half.com pay periods start and end at midnight Pacific time, but the time values are stored in the eBay database in GMT (not Pacific time). See "[Time Values](#)" on page 960 for information about converting between GMT and Pacific time.
- Your current pay period only contains pending payments. Your most recent pay period may also contain pending payments until the payments are settled. You can retrieve pending payments for any single pay period.
- Your current pay period never contains paid payments. You can retrieve paid payments for a single pay period approximately a week after the period ends.
- When you use [GetSellerPayments](#), you need to specify which pay period you are interested in. To do this, you specify a time range in the request. eBay determines which pay period best matches the time range you specified, and retrieves data from that pay period.

To retrieve paid payments, always specify a time range that starts before the pay period and ends after the pay period. Otherwise, you will not get any data. For example, if your pay period starts on 2005-09-16 and ends on 2005-09-30, your time range (in GMT) could start 8 hours before

the beginning of the period (2005-09-16T00:00:00.000Z) and end 4 hours after the end of the period (2005-10-01T12:00:00.000Z).

When you retrieve pending payments, you can specify a smaller time range. However, eBay may also return additional pending payments within the same period. If you need to refine the results further, you can filter the data locally.

If you specify a time range that fully covers two pay periods, only the payments from the most recent pay period are returned. The earliest time you can specify is 18 months in the past (as older data is not returned).

- As a given order line item could result in both a sale and (later) a refund, it is possible to retrieve multiple payments for the same order line item (one for the sale and one for the refund). Therefore, you should not assume that a transaction ID will only appear once in your payment history. You may be able to use the Item ID/Transaction ID pair (or OrderLineItemID) and paid date (and sale or refund payment type) if you need to uniquely identify payments.

Applicable Fields for Half.com Listings

Half.com sellers typically prefer to list many items at the same time. To facilitate bulk listing, only a subset of eBay's **Item** fields are applicable to Half.com listings. In addition, there are some special **Item** fields that are *only* applicable to Half.com listings.

In general, when you list Half.com items, you should only use fields that are flagged as applicable to Half.com. The **Item** fields shown in [Table 36-1](#) are most commonly used for Half.com listings.

These fields and their usage are described in more detail in the eBay Trading API Call Reference for [AddItem](#). The eBay Trading API Call Reference may identify additional fields that are also applicable to Half.com.

For information about the calls applicable to Half.com, see "[Applicable Calls for Half.com](#)" on page 579.

You can find information about [GetFeedback](#) and [LeaveFeedback](#) by following links in the "[Overview of Calls](#)" on page 770.

For information about the main eBay eBay Trading API Call Reference document, see "[Schema Reference Documentation](#)" on page 967.

Table 36-1 Commonly Used AddItem Fields for Half.com Listings

AddItem Input Field	Brief Summary
InvocationID	A unique identifier for a particular call. Use this to avoid submitting the same listing twice. If the same InvocationID is passed in after it has been passed in previously for a call that succeeded for a particular application and user, an error will be returned.

Table 36-1 Commonly Used AddItem Fields for Half.com Listings

AddItem Input Field	Brief Summary
<code>Item.AttributeArray. Attribute [attribute attributeLabel]</code>	<p>Use this XML attribute to identify that you are passing the item condition or notes to the buyer.</p> <p>Possible values of the <code>attributeLabel</code> XML attribute:</p> <ul style="list-style-type: none"> ■ Condition ■ Notes <p>(Specify the complete <code>Attribute</code> property twice in the <code>AttributeArray</code>—one for the condition and one for the notes.)</p> <p>Note: This property does not use eBay's standard Item Specifics (Attributes) API. That is, this uses a different format than <code>Item.AttributeSetArray</code>.</p> <p>In <code>GetItem</code>, the condition is returned in <code>Item.AttributeSetArray</code> instead.</p>
<code>Item.AttributeArray. Attribute.Value. ValueLiteral</code>	<p>Use this field to specify the item condition or notes to the buyer.</p> <p>If <code>attributeLabel = Condition</code>, you can pass one of these values:</p> <ul style="list-style-type: none"> ■ <code>Brand_New</code> ■ <code>Like_New</code> ■ <code>Very_Good</code> ■ <code>Good</code> ■ <code>Acceptable</code> <p>If <code>attributeLabel = Notes</code> (for a note to the buyer) you can pass a string that contains a maximum of 500 characters. The string cannot include HTML markup.</p> <p>Note: This property does not use eBay's standard Item Specifics (Attributes) API. That is, this uses a different format than <code>Item.AttributeSetArray</code>.</p>
<code>Item.Country</code>	The country where the item is located. Specify <code>US</code> .
<code>Item.Currency</code>	The currency associated with the price. Specify <code>USD</code> . Half.com is only available in the United States.
<code>Item.ExternalProductID</code>	An ISBN or UPC value and related meta-data. All Half.com items are listed by using data that is pre-filled from the Half.com catalog. When you list an item, you specify its ISBN or UPC value, and Half.com fills in the category, description, stock photo, and Item Specifics for you. Note: This means you should <i>not</i> specify <code>PrimaryCategory</code> , <code>Description</code> , <code>PictureURL</code> , or eBay Item Specifics in your listing request.
<code>Item.ListingDuration</code>	Half.com listings can have an unlimited duration called Good 'Til Canceled (GTC).
<code>Item.ListingType</code>	Specify <code>Half</code> for all Half.com listings.
<code>Item.Location</code>	Specify the location of the item.
<code>Item.PrivateNotes</code>	Comments that will be visible only to the seller.
<code>Item.Quantity</code>	The number of items that can be purchased from the listing.
<code>Item.SellerInventoryID</code>	An identifier (may be a SKU) that is unique across the seller's active (unsold) inventory. Note that for eBay.com listings, <code>Item.SKU</code> can be used for a seller's SKU; see "Using a SKU Identifier with Items" on page 239.
<code>Item.StartPrice</code>	The price of each item in the listing, in US Dollars. (All items in the same listing are the same price.) Also, specify <code>USD</code> for this element's <code>currency</code> XML attribute.
<code>RequesterCredentials. eBayAuthToken</code>	As with all calls, specify the seller's authentication token. See "Security" on page 101.
<code>Version</code>	Specify the version of the schema you are using.

Most of the other fields that would normally be specified in **AddItem** (buyer payment methods, shipping costs, etc.) are not specified for individual listings. Instead, either Half.com determines the data (e.g., flat shipping rates) or the seller specifies their preferences for all listings in their Half.com account. If you pass such details in **AddItem**, they are ignored. In some cases, errors may occur if you specify inapplicable fields when you list an item.

The table below indicates which **AddItem** fields can be revised by using **Reviseltem**. The meaning and format of each field is the same as the **AddItem** format. For information about using **Reviseltem**, including the special fields you need to pass in to identify which fields you are modifying, see “[Revising Items](#)” on page 381. You can only revise Half.com listings that contain active (unsold) items.

Table 36-2

Field	Summary
<code>Item.AttributeArray. Attribute [attribute attributeLabel]</code>	Use this XML attribute to identify that you are passing the item condition or notes to the buyer.
<code>Item.AttributeArray. Attribute.Value. ValueLiteral</code>	Use this field to modify the item condition or notes to the buyer.
<code>Item.ItemID</code>	You cannot modify the item ID. However, to identify the item you want to modify, you need to pass in either the <code>ItemID</code> (which was returned after using AddItem) or <code>SellerInventoryID</code> (if you specified this as input when you used AddItem).
<code>Item.ListingType</code>	Specify <code>Half</code> for all Half.com listings, otherwise the correct logic won't be used. If you don't specify this value, Reviseltem could return strange errors, or result in unwanted side effects on the item (even if no errors are returned).
<code>Item.PrivateNotes</code>	Use this to modify the comments that will be visible only to the seller.
<code>Item.Quantity</code>	Use this to modify the number of items that can be purchased from the listing.
<code>Item.SellerInventoryID</code>	You cannot modify the seller inventory ID. However, to identify the item you want to modify, you need to pass in either the <code>ItemID</code> (which was returned after using AddItem) or <code>SellerInventoryID</code> (if you specified this as input when you used AddItem).
<code>Item.StartPrice</code>	Use this to modify the price of each item in the listing, in US Dollars. (All items in the same listing are the same price.)

When you use **AddItem**, the response includes the Half.com item ID, which you can use to identify the item when you make other API calls (e.g., **Reviseltem** or **EndItem**). You can ignore the start and end time of the item and eBay listing fees. They are standard results for the **AddItem** family of calls, but they are not applicable to Half.com listings.

Applicable Calls for Half.com

Due to the way Half.com data is stored, and due to differences between Half.com and eBay, a subset of the eBay API calls are available for Half.com. In addition, there are some calls that are exclusively used for Half.com.

The three tables below provide a summary of the calls applicable to Half.com.

Please refer to the standard Trading API Call Reference for each call. That standard Trading API Call Reference indicates which fields of each call can apply to Half.com.

[Trading API Call Reference](#)

In [Table 36-3](#), the calls that you can use to manage your Half.com inventory are listed. You must be a registered Half.com seller in order to use these calls for Half.com listings.

Table 36-3

Call	Summary
AddItem	Use this call to list items to Half.com.
EndItem	Use this call to end a Half.com listing before all items have sold.
EndItems	Use this call to end multiple Half.com listings before all items have sold.
ReviseItem	Use this call to modify one of your active (unsold) listings.
VerifyAddItem	Use this to test your AddItem data before you list Half.com items.
GetItem	<p>Use this to retrieve data about a Half.com listing. (Because Half.com sellers tend to have very large inventories, calling GetItem may not be practical for Half.com sellers.)</p> <p>As Half.com categories and Item Specifics are different from eBay's, GetItem uses some workarounds to map the data:</p> <p>The Half.com condition value Brand_New is mapped to the eBay condition value New, and all other Half.com condition values are mapped to Used.</p> <p>We return the original Half.com condition and the mapped eBay condition in Item.AttributeSetArray. This is different from the label/value format you use to specify the Half.com condition in AddItem. If you are used to working with attributes for eBay items, please note that you can use GetAttributesCS to look up the eBay condition IDs, but not the Half.com condition IDs.</p> <p>GetItem doesn't return Half.com catalog product details for Half.com listings.</p>

[Table 36-4](#) lists the calls you can use to manage your Half.com orders (items that have been purchased) and payments. You must be a registered Half.com seller in order to use these calls for Half.com listings.

Table 36-4

Call	Summary
CompleteSale	Marks items complete. This includes the ability to mark items Paid or Shipped, and to leave feedback.
GetItemTransactions	<p>Retrieves order line item information for a specified item ID. Depending on whether the listing sells one or multiple items and on whether anything has been sold in that listing, this returns zero, one, or multiple order line items.</p> <p>A order line item represents a purchasing relationship between one seller, one buyer, and one listing. On eBay, some listings can contain multiple items (Quantity >1), so a order line item can include multiple items from the same listing, all purchased at the same time (with a single click of the Buy It Now button).</p>

Table 36-4

Call	Summary
GetOrders	Retrieves single and multiple line item orders. An order consists of one or more order line items involving the same buyer and seller. A seller typically creates a Combined Payment order when they want to combine two purchases from the same buyer into one package to reduce shopping costs. This call retrieves all orders for which the requesting user is a participant (as a buyer or seller).
GetOrderTransactions	Retrieves information about one or more orders or one or more order line items (or both).
GetSellerPayments	Retrieves information about your pending and paid payments for your Half.com sales. This also provides payment information related to refunds you have issued to buyers.
GetSellerTransactions	Retrieves order line item information for the user for which the call is made (and not for any other user), where an order line item is the information about the sale of one or more items by one buyer from a single listing.
IssueRefund	Use this call to issue a refund to a buyer for a single order line item.

Table 36-5 lists calls that provide information about eBay and Half.com users.

Table 36-5

Call	Summary
GetFeedback	Use this call to retrieve a user's feedback.
GetUser	Use this call to retrieve information about a user. Please note that this call does not provide information that is specific to Half.com.
LeaveFeedback	Use this call to leave feedback for a user.

Certain other eBay calls may also be of interest to Half.com sellers. However, calls other than the ones described above do not provide information that is specific to Half.com.

Half.com does not support calls that retrieve order line item data in bulk, retrieve category meta-data, manage disputes or unpaid items, or perform other tasks that are specific to listings on the main eBay.com site.

To verify the details of the items a seller listed, the seller can use eBay's File Exchange application to retrieve their active listings.

File Exchange
http://pages.ebay.com/file_exchange/

Other Services and Calls Supporting Half

The Shopping API provides the FindHalfProducts call, which searches Half.com for stock product information (stock description and Item Specifics), such as information about a particular kind of DVD or book. The call also, retrieves up to 30 Half.com listings associated with a product.

For more information, see the [eBay Developer documentation home page](#).

Chapter 37

International Differences Overview

If your application works with items on sites outside the US, or if your application can be used with more than one site, you should understand the differences in the validation rules, business logic, and overall usage rules that are applied for different eBay sites.

The following information focuses on site differences by country. For information about other types of sites, such as the US eBay Motors site, see these topics:

- “[eBay Sites and Environments](#)” on page 149
- “[Working with Special eBay Sites](#)” on page 557

As described in “[The eBay Developers Program Release Cycle](#)” on page 85, eBay releases applicable functionality to international sites (e.g., ebay.co.uk) about one week after the functionality is released to US sites (e.g., eBay.com). This means the features described in this section will be available about one week after the Version 817 release.

The API supports calls for international sites through eBay's Sandbox or Production SOAP Gateway (see “[Routing the Request \(Gateway URLs\)](#)” on page 87). The Gateway routes calls for international sites to eBay's international pool.

The international pool enforces the logic differences between the US and international sites. Examples of these differences are listed below. Additional differences are summarized in “[Field Differences for eBay Sites](#)” on page 590.

- Checkout is available to all eBay sites, including International sites. However, some Checkout-related data elements are only used on the US site. These US-only tags include:
 - [Item.ShippingDetails.SalesTax.SalesTaxPercent](#)
 - [Item.ShippingDetails.SalesTax.SalesTaxState](#)
 - [Item.ShippingDetails.SalesTax.ShippingIncludedInTax](#)
- Available payment methods differ some from one eBay site to the next. See “[Payment Method Differences \(PaymentMethod\)](#)” on page 601 for a chart containing valid payment methods for each site.
- Different international sites use different sales tax fields. See “[Tax Field Differences \(SalesTax, UseTaxTable\)](#)” on page 614 for a chart containing valid tax fields for all international sites.
- See “[Email Privacy Policy](#)” on page 524.
- Certain countries have regulations governing business sellers or business-to-business (B2B) transactions. A seller that represents a business (a *business seller*) can use eBay's Business User

features to offer an item exclusively to bidders and buyers that also represent businesses. See also “[Working with Business Features and VAT](#)” on page 586.

- Some countries, such as the United Kingdom, Ireland, Belgium (Dutch and French), and France now require sellers to change their account status from ‘personal’ to ‘business’ after a certain threshold of trading has been exceeded, in order to comply with EU law.
- Business sellers are required to accept returns for Buy It Now items in specific countries, such as the United Kingdom, Ireland, Germany, Austria, and Switzerland.
- Some eBay users are subject to Value-Added Tax (VAT). See “[Working with Business Features and VAT](#)” on page 586.
- Charity or fundraising listings are supported on the US and Motors sites (eBay Giving Works) and UK site (eBay for Charity). For more information, see “[Identifying Listings that Benefit Nonprofits](#)” on page 234.
- The Immediate Payment feature is only applicable for qualifying listings on the US and UK sites.
- CIP and CIP+ are only applicable for listings on the Germany (DE), Austria (AT), and Switzerland (CH) sites.
- On the India site (site code 203), the sale of items in some categories, such as Movies and Videos, requires pre-approval. In addition, an international listing fee applies when the seller lists items on the India site that are available for shipping outside of India. For more information, see the eBay Help on the India site.

In addition, you should be aware of the call differences and values an application must use so that the call works properly for international users and applications. See “[Working with Internationalization Issues](#)” on page 584.

Chapter 38

Working with Internationalization Issues

This topic addresses some of the issues for the eBay API concerning its use outside the US and with languages in different character sets. For a list of valid fields and values for listing and searching for items, see “[Field Differences for eBay Sites](#)” on page 590.

“[Internationalization of the API](#)” on page 584

“[Effects of Site IDs](#)” on page 585

Internationalization of the API

This section discusses some of the call differences and values an application must use so that the call works properly for international users and applications. Detailed changes that pertain to a particular API call are described in the topic for the call. There are a number of changes that are more global in nature.

Site ID: This URL query string parameter (see “[Routing the Request \(Gateway URLs\)](#)” on page 87) gives an indication of the eBay site on which the item will be listed. See “[Effects of Site IDs](#)” on page 585 for the effect of this parameter.

Error strings: Return error strings in the appropriate language for a site by using the `ErrorLanguage` property of a call’s request object. `ErrorLanguage` can be used with any API call. Use the same value for Site ID as the value for `ErrorLanguage` to return the error strings in the corresponding language.

Translation of non-user data: Certain pieces of data returned by API calls originate with eBay and are not entered by any end-user. This includes such things as the default names for eBay Stores categories that have not been renamed. These pieces of information that do not originate as user-entered data are returned in the language indicated by the Site ID URL query string parameter. User-entered data is returned as it was entered.

Other business logic: Other factors or behaviors that would change from one international locale to the next are addressed within the documentation of each call on an as-needed basis. Details about differences in the business logic of individual fields and code lists are summarized in “[Field Differences for eBay Sites](#)” on page 590.

Currency symbols and IDs: All supported currencies are listed in the code list `CurrencyCodeType`. The currency IDs in this code list that have descriptions are used as input for calls associated with listing items (like `AddItem`). The currency IDs and symbols are returned as fields by calls like `GetItem`. Currencies are associated with items, not the site ID of the function requestor.

Effects of Site IDs

Items are listed on a specific site and users are registered with a specific site. When calling a function, you must specify a value for the Site ID URL query string parameter (see “[Routing the Request \(Gateway URLs\)](#)” on page 87) so that the call is properly associated with a site.

The Site ID parameter is used to give an indication of the locale associated with the request being made. This information (and, in some cases, the user's registered site ID) impacts such things as international effects on business rules (e.g., support for or applicability of certain data input properties), the translation of error strings, and the translation of non-user data returned by calls (e.g., item prices). It can also affect how data that appears on the eBay site is localized (e.g., time stamps). See “[Specifying the Target Site](#)” on page 93 for more information about these effects.

Character Encoding

The API only supports the UTF-8 encoding scheme. As described in “[Summary of Data Types](#)” on page 959, all characters in the Unicode character set are supported.

Chapter 39

Working with Business Features and VAT

This chapter covers Value-Added Tax and the Business-to-Business feature.

- “[Value-Added Tax \(VAT\)](#)” on page 586
- “[Adding a Business Item](#)” on page 588
- “[Modifying Business Items](#)” on page 588
- “[Retrieving Information about Business Users and Items](#)” on page 589

Value-Added Tax (VAT)

The following information applies to listings on VAT-enabled sites. On a VAT-enabled site, you can specify a VAT percent value for a listing. For information on which sites are VAT-enabled, see “[Business Feature Field Differences](#)” on page 591.

Business sellers in Germany (DE), Austria (AT), or Switzerland (CH) have additional business-to-business (B2B) options that are discussed in “[Business Features on German Sites](#)” on page 587.

Value-Added Tax (VAT) is a tax on services, including services provided by eBay. You should consult an independent tax advisor and/or a local tax authority to determine what laws and rules apply to you, including whether you should register for a VAT ID with eBay and whether, and how much, VAT must be charged on items you sell.

The VAT rate varies depending on where you reside (or have a permanent address or establishment). You, or the sellers who use your application, should ensure that the registration address you provide to eBay is correct, especially the country designation.

Note: In general, on VAT-enabled sites, the ListingFee value that results from listing an item is the gross amount, meaning it includes VAT. However, on the India site, the ListingFee value is the net amount, meaning it does not include VAT.

VAT-exempt Sellers

You can avoid paying VAT charges on your eBay fees by registering a valid VAT-ID with eBay. You are then *VAT-exempt* with respect to eBay and will not be required to pay VAT charges on your eBay fees. You are responsible for paying the required VAT taxes on the goods you sell.

Your VAT status—exempt or non-exempt—is recorded in the `User.VATStatus` property returned from several calls. Of particular interest are these two values from the `VATStatusCodeType` code list:

- `VATTax`—user is subject to VAT charges on eBay fees
- `VATExempt`—user is registered with eBay as VAT exempt

The following calls retrieve information about a user's VAT status in the response object:

[GetItem](#)
[GetItemTransactions](#)
[GetSellerList](#)
[GetSellerTransactions](#)
 [GetUser](#)

Advantages of Being VAT-exempt

As an eBay VAT-exempt seller, you have the following advantages:

- You do not pay VAT charges on your eBay fees.
- You can specify a [VATPercent](#) between zero and 30 percent for any item that you list on a VAT-enabled site. This VAT information is visible to potential buyers on the item's listing page.
- When you provide a [VATPercent](#), you have the option of choosing an invoice template that includes the net price (the basic price without VAT), the VAT percent, the VAT amount, and the gross price (net price plus VAT).

Business Features on German Sites

eBay sites in Germany, Austria, and Switzerland offer German, Austrian, and Swiss users the option of being a business-to-business (B2B) seller who offers designated items, called Business Items, only to buyers who confirm that they are business users and are buying the item for business purposes. Private users – users who do not confirm that they represent businesses – cannot bid on business items.

To be a B2B seller, you must register a valid VAT-ID with eBay. (See "[VAT-exempt Sellers](#)" on page 586.)

Advantages of Being a B2B Seller

When you are a B2B seller listing a Business Item, a message appears at the bottom of the item's listing page requiring potential bidders to confirm that they are business buyers. Note that you are responsible for confirming a buyer's business status. eBay does not verify such declarations.

In addition, if you specify a [VATPercent](#) for an item, your VAT rate is displayed on the item's listing page and you can choose to create an invoice that displays net price, VAT rate, and gross price, in accordance with European law.

Adding a Business Item

To offer a B2B item, you must meet the following conditions:

- You are eBay VAT-exempt.
- You live in Germany, Austria or Switzerland.
- You are listing the item on a site in Germany, Austria or Switzerland.
- You are offering the item only in B2B-enabled categories.

To check whether a category is B2B-enabled, check the value of the `Category.B2BVATEnabled` property in the category data returned by `GetCategories`.

To list a business item using the `AddItem`:

- Use `GetCategories` to make sure the categories in which the item will be listed are B2B-enabled (i.e., `Category.B2BVATEnabled` returns true).
- Pass a value of true in the `Item.VATDetails.BusinessSeller` property. This means the seller declares that the seller represents a business.
- Pass a value of true in the `Item.VATDetails.RestrictedToBusiness` argument. This means the seller restricts the sale to another business user.
- (*Optional*) Specify a value between 0 and 30 percent in the `Item.VATDetails.VATPercent` property and include the VAT in the starting price (minimum bid). This displays VAT information in the item's shipping page and allows you to choose an invoice template that displays the net price, VAT rate and amount, and the gross price.
- Some countries require that you specify a return policy for any Buy It Now items.

Regardless of whether an item uses business features or not, eBay fees are charged on gross sales prices.

Usage of some business features is optional. For more information about eBay's business features, see the eBay online help:

Business Features
<http://pages.ebay.de/help/sell/Businessfeatures.html>

Modifying Business Items

When revising or relisting an item that was previously listed as a business item, you do not need to supply values for the B2B properties (`VATDetails.BusinessSeller`, `VATDetails.RestrictedToBusiness`, or `VATDetails.VATPercent`) unless you want to modify their values.

An item will no longer be qualified as a business item (i.e., the listing will no longer qualify for business features and will be converted to a non-business listing) if any of the following conditions are true when the item is revised or relisted:

- The seller's country of residence has changed to an ineligible country.
- The seller has changed the categories of the item being re-listed to one or more that do not support business features.
- The seller sets either `VATDetails.BusinessSeller` or `VATDetails.RestrictedToBusiness` to false.

If an item was not qualified as a business item when originally listed, but meets the conditions specified under “[Adding a Business Item](#)” on page 588 above when the item is revised or relisted, the seller can convert the item to a business item by specifying the B2B arguments.

Retrieving Information about Business Users and Items

Retrieve the item data with `GetItem` and examine the `Item.VATDetails.RestrictedToBusiness` property and the `Item.VATDetails.VATPercent` field to determine whether an item was listed as a business item and whether the seller chose to specify the VAT rate.

Chapter 40

Field Differences for eBay Sites

This chapter describes some of the differences among the eBay sites. Where applicable, the sections in this chapter show differences in the data format that these fields accept or return. Fields and codes not shown in this chapter are equally supported across all eBay sites.

Each table below lists the eBay site abbreviation (e.g., UK) and the corresponding site ID (e.g., 3). See the site code list ([SiteCodeType](#)) for explanations of the abbreviations that are used for each site (e.g., UK is United Kingdom) or see the site ID table below. Also see “[International Differences Overview](#)” on page 582.

- “[Business Feature Field Differences](#)” on page 591
- “[Buyer Requirements, Verified User \(VerifiedUser\)](#)” on page 591
- “[Currency Codes](#)” on page 593
- “[Fees per Site](#)” on page 593
- “[Hit Counter Differences \(HitCounter\)](#)” on page 595
- “[Listing Duration Logic Differences](#)” on page 596
- “[Minimum Feedback Requirements for Various Features](#)” on page 597
- “[Miscellaneous Item Field Differences](#)” on page 600
- “[Payment Method Differences \(PaymentMethod\)](#)” on page 601
- “[Postal and Zip Codes \(PostalCode\)](#)” on page 603
- “[Real Estate Ad Format Differences](#)” on page 604
- “[Search Type Differences](#)” on page 605
- “[Seller Level Differences \(SellerLevel\)](#)” on page 607
- “[Seller Payment Methods \(SellerPaymentMethod\)](#)” on page 608
- “[Shipping Region and Shipping Option Differences](#)” on page 609
- “[Promotional Sale Differences](#)” on page 610
- “[Site IDs](#)” on page 611
- “[Site ID Filters for Search Calls \(SiteIdFilter\)](#)” on page 613
- “[Tax Field Differences \(SalesTax, UseTaxTable\)](#)” on page 614
- “[Tax Jurisdictions](#)” on page 615

Business Feature Field Differences

The table below describes Business Feature field differences for various sites. See “[Working with Business Features and VAT](#)” on page 586.

Table 40-1 Business Feature Differences

Site (and ID)	B2BVATEnabled	BusinessSeller	RestrictedToBusiness	VATPercent
AT (16)	yes	yes	yes	yes
AU (15)	no	no	no	no
BEFR (23)	no	yes	no	yes
BENL (123)	no	yes	no	yes
CA (2)	no	no	no	no
CAFR (210)	no	no	no	no
CH (193)	yes	yes	yes	yes
DE (77)	yes	yes	yes	yes
ES (186)	no	no	no	yes
FR (71)	no	yes	no	yes
HK (201)	no	no	no	no
IE (205)	no	yes	no	yes
IN (203)	no	yes	no	yes
IT (101)	no	no	no	yes
MY (207)	no	no	no	no
NL (146)	no	no	no	yes
PH (211)	no	no	no	no
PL (212)	no	no	no	no
SG (216)	no	no	no	no
UK (3)	no	yes	no	yes
US (0)	no	no	no	no

Buyer Requirements, Verified User (VerifiedUser)

`Item.BuyerRequirementDetails` is part of a mechanism available to sellers to reduce the likelihood of receiving fraudulent bids. A sub-feature of this is `Item.BuyerRequirementDetails.VerifiedUserRequirements.VerifiedUser`.

(Please note that “verified” in this context does not mean that eBay is certifying or not certifying a particular user. Rather, this is a simply a rough check on another bidder attribute for the sake of the seller.)

As part of this mechanism, eBay may check (depending on the site) to see which of the following is true for a bidder:

- 1 has a PayPal account
- 2 has a credit card on file
- 3 has completed phone verification (via Authentify.com)
- 4 has completed SMS verification

These are listed in order of significance.

If a seller sets **VerifiedUser** to true for a listing and if a bid is made by a bidder registered with the same site, eBay checks the bidder for the minimum requirement noted in the table below. (Exception: “N/A” for a site in the table below means that a seller registered on that site is not given the opportunity to block bidding via **VerifiedUser**.) If the user does not meet that minimum requirement, eBay works upward through the list above until a match is found. (It is irrelevant, for this check, whether any of the items lower in the list are true.) If a match is not found, that bidder is blocked from bidding. (A blocked bidder can contact the seller and request to be exempted from the requirement.)

If the seller and bidder are from different registration sites, the table below is not used. Instead, eBay checks whether the bidder has a credit card on file. If the user does not, the bidder is presented with the option to enter credit card information and can continue with bidding. (Exception: no such option is presented to bidders registered on these sites: IN, IT, US, eBay Motors. The bidder is simply blocked from bidding.)

Table 40-2 Minimum Requirements, VerifiedUser

Site (and ID)	Credit card is on file	Went through phone verification	Went through SMS verification
AT (16)	N/A	N/A	N/A
AU (15)	no	yes	no
BEFR (23)	N/A	N/A	N/A
BENL (123)	N/A	N/A	N/A
CA (2)	N/A	N/A	N/A
CH (193)	N/A	N/A	N/A

Table 40-2 Minimum Requirements, VerifiedUser (Continued)

Site (and ID)	Credit card is on file	Went through phone verification	Went through SMS verification
DE (77)	N/A	N/A	N/A
ES (186)	no	yes	no
FR (71)	no	yes	no
HK (201)	N/A	N/A	N/A
IE (205)	yes	no	no
IN (203)	no	no	yes
IT (101)	N/A	N/A	N/A
MY (207)	N/A	N/A	N/A
NL (146)	N/A	N/A	N/A
PH (211)	N/A	N/A	N/A
PL (212)	N/A	N/A	N/A
SG (216)	N/A	N/A	N/A
UK (3)	yes	no	no
US (0)	N/A	N/A	N/A
eBay Motors (100)	N/A	N/A	N/A

Currency Codes

For a list of all supported currency codes, use [GeteBayDetails](#).

For a list of which currencies can be specified as input to listing calls like [AddItem](#) for a given site, see the annotations in [CurrencyCodeType](#) and/or [SiteCodeType](#).

Be aware that [GetItem](#) will return the item's original currency, regardless of which site you send your [GetItem](#) request to.

Fees per Site

Not all sites have a page that displays fee information.

Table 40-3 Fees per Site

Site (and ID)	URL for Fees
AT (14)	http://pages.ebay.at/help/sell/fees.html
AU (15)	http://pages.ebay.com.au/help/sell/fees.html
BEFR (23)	http://www.befr.ebay.be/help/sell/fees.html
BENR (123)	http://www.benl.ebay.be/help/sell/fees.html
CA (2)	http://pages.ebay.ca/help/sell/fees.html
CAF'R (210)	http://pages.cafr.ebay.ca/help/sell/fees.html
CH (193)	http://pages.ebay.ch/help/sell/fees.html
DE (77)	http://pages.ebay.de/help/sell/fees.html
ES (186)	http://pages.ebay.es/help/pay_fees/fees.html
FR (71)	http://pages.ebay.fr/help/sell/fees.html
HK (201)	http://pages.ebay.com.hk/help/sell/fees.html
IE (205)	http://pages.ebay.ie/help/sell/fees.html
IN (203)	http://pages.ebay.in/help/account/fees.html
IT (101)	http://pages.ebay.it/help/sell/fees.html
MY (207)	http://pages.ebay.com.my/help/sell/fees.html
NL (146)	http://pages.ebay.nl/help/sell/fees.html
PH (211)	http://pages.ebay.ph/help/sell/fees.html
PL (212)	http://pages.ebay.pl/help/sell/fees.html
SG (216)	http://pages.ebay.com.sg/help/sell/fees.html
UK (3)	http://pages.ebay.co.uk/help/sell/fees.html
US (0)	http://pages.ebay.com/help/sell/fees.html
eBay Motors	http://pages.ebay.com/help/sell/motorfees.html
eBay Stores	http://pages.ebay.com/help/sell/storefees.html
eBay Real Estate	http://pages.ebay.com/help/sell/realestatefees.html

Hit Counter Differences (HitCounter)

The US is the only site supporting the **Hidden**, **HiddenStyle**, **HonestyStyle**, and **GreenLED** counter types.

Table 40-4 Hit Counter Code Differences

Site (and ID)	NoHitCounter	BasicStyle	RetroStyle	HiddenStyle	HonestyStyle	GreenLED	Hidden
AT (16)	yes	yes	yes	no	no	no	no
AU (15)	yes	yes	yes	no	no	no	no
BEFR (23)	yes	yes	yes	no	no	no	no
BENL (123)	yes	yes	yes	no	no	no	no
CA (2)	yes	yes	yes	no	no	no	no
CAF'R (210)	yes	yes	yes	no	no	no	no
CH (193)	yes	yes	yes	no	no	no	no
DE (77)	yes	yes	yes	no	no	no	no
ES (186)	yes	yes	yes	no	no	no	no
FR (71)	yes	yes	yes	no	no	no	no
HK (201)	yes	yes	yes	no	no	no	no
IE (205)	yes	yes	yes	no	no	no	no
IN (203)	yes	yes	yes	no	no	no	no
IT (101)	yes	yes	yes	no	no	no	no
MY (207)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
NL (146)	yes	yes	yes	no	no	no	no
PH (211)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
PL (212)	yes	yes	yes	no	no	no	no
SG (216)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
UK (3)	yes	yes	yes	no	no	no	no
US (0)	yes	yes	yes	yes	yes	yes	yes

Listing Duration Logic Differences

Notes:

- For 1-day listing seller requirements, see “[Minimum Feedback Requirements for Various Features](#)” on page 597.
- US and Canada sellers must pay an additional fee for 10-day duration listings.
- For vehicle listings on the US eBay Motors site and the Canadian site, 1-day listing is not allowed. The 1-day listing option is not available for most Motor categories of the following sites: AU, BEFR, BENL, IN, IT, NL.

Important: The following table is outdated. The available information will remain in the documentation to help developers whose applications do not yet support [GetCategoryFeatures](#). However, we recommend that you start using [GetCategoryFeatures](#) to determine which listing formats support each duration on the eBay site you are using. See “[GetCategoryFeatures](#)” on page 808. Also see this *Knowledge Base* article:

How to find valid ListingDurations for the different types of items on each site
https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=426

Table 40-5 Listing Duration Logic Differences

Site (and ID)	Auction Listing Durations	Buy It Now Only! Listing Durations	Ad Format Listing Durations	Store Inventory Format (deprecated) Listing Durations
AT (16)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	no	30, GTC
AU (15)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	no	30, GTC
BEFR (23)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	no	30, GTC
BENL (123)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	no	30, GTC
CA (2)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	no	30, GTC
CH (193)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	no	30, GTC
DE (77)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	10, 30	30, GTC
ES (186)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	no	30, GTC
FR (71)	3, 5, 7, 10	3, 5, 7, 10	no	30, GTC

Table 40-5 Listing Duration Logic Differences (Continued)

Site (and ID)	Auction Listing Durations	Buy It Now Only! Listing Durations	Ad Format Listing Durations	Store Inventory Format (deprecated) Listing Durations
HK (201)	3, 5, 7, 10	3, 5, 7, 10	no	N/A
IE (205)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	no	30, GTC
IN (203)	1, 3, 5, 7, 10	3, 5, 7, 10	30, 90	30, GTC
IT (101)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	no	30, GTC
MY (207)	1, 3, 5, 7, 10	3, 5, 7, 10	30	30, 60, 90, GTC
NL (146)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	no	30, GTC
PH (211)	1, 3, 5, 7, 10	3, 5, 7, 10	30	30, 60, 90, GTC
PL (212)	1, 3, 5, 7, 10	3, 5, 7	no	30, GTC
SG (216)	3, 5, 7, 10	3, 5, 7, 10	30	N/A
UK (3)	1, 3, 5, 7, 10	3, 5, 7, 10, 20, 30	no	30, GTC
US (0)	1, 3, 5, 7, 10	none	30, 90	30, GTC

Minimum Feedback Requirements for Various Features

A user's feedback score provides a gauge that prospective buyers and sellers can use to assess the user's reputation on the eBay site. The feedback score also can be a factor in enabling eBay users to take advantage of various features and levels of participation in those features. For example, a user must have a certain feedback score to open an eBay store. (On some sites, additional requirements must also be met in addition to the feedback score.)

The table below lists the required feedback score necessary to use certain features. These requirements vary by site.

Minimum feedback requirements are lowered for some features when PayPal is offered as a payment method.

The table below uses these conventions:

- FWP = Feedback with PayPal
- FWoP = Feedback without PayPal
- FNA = Feature not available on that site
- N/A = Feedback is not a requirement for the feature on the specified site

Table 40-6 Feedback Requirements With and Without PayPal

Site (and ID)	Create A Store (Private Seller)	Create A Store (Business Seller)	List Fixed Price Item	Add Buy It Now Price To An Auction	Select 1-Day Duration	Specify Quantity For Auction	Specify Quantity >1 For Fixed Price	Featured Plus	Home Page Featured
AT (16) FWP	10	0	0	10	10	20	20	10	FNA
AT (16) FWoP	10	0	0	10 or ID verified	10	20	20	10	FNA
AU (15) FWP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
AU (15) FWoP	5	5	5	5	5	5	20	10	10
BEFR (23) FWP	5	5	5	5	5	10	10	10	10
BEFR (23) FWoP	10	10	10	10	10	20	20	10	10
BENL (123) FWP	3	3	3	3	3	10	10	10	10
BENL (123) FWoP	5	5	5	5	5	20	20	10	10
CA (2) FWP	0	0	0	5	5	15	15	10	FNA
CA (2) FWoP	10	10	0	10	10	30	30	10	FNA
CAFR (210) FWP	0	0	0	5	5	15	15	10	FNA
CAFR (210) FWoP	10	10	0	10	10	30	30	10	FNA
CH (193) FWP	10	0	0	10	10	20	20	10	FNA
CH (193) FWoP	10 or ID verified	0	0	10 or ID verified	10	20	20	10	FNA
DE (77) FWP	10	0	0	10	10	20	20	10	10
DE (77) FWoP	10	0	0	10 or ID verified	10	20	20	10	10

Table 40-6 Feedback Requirements With and Without PayPal (Continued)

Site (and ID)	Create A Store (Private Seller)	Create A Store (Business Seller)	List Fixed Price Item	Add Buy It Now Price To An Auction	Select 1-Day Duration	Specify Quantity >1 for Auction	Specify Quantity >1 for Fixed Price	Featured Plus	Home Page Featured
ES (186) FWP	10	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
ES (186) FWoP	10 or Direct Debit on File	10 or Direct Debit on File	5	5 or Direct Debit on File	5	20	20	10	10
FR (71) FWP	5	0	5	5	5	10	10	N/A	10
FR (71) FWoP	10	0	10	10 or Direct Debit on File	10	20	20	N/A	10
HK (201) FWP	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A
HK (201) FWoP	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A
IE (205) FWP	5	0		5	5	10	10	10	10
IE (205) FWoP	10	0	0	10 or Direct Debit on file	10	20	20	10	10
IN (203) FWP	5	5	0	5	5	20	20	10	10
IN (203) FWoP	5	5	0	5	5	20	20	10	10
IT (101) FWP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
IT (101) FWoP	10	10	10	5 or Direct Debit on File	5	5	20	20	10
MY (207) FWP	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A
MY (207) FWoP	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A
NL (146) FWP	3 or Direct Debit on File	3 or Direct Debit on File	3	3	3	10	10	10	10
NL (146) FWoP	5	5		5 or Direct Debit on File	5	20	20	10	10
PH (211) FWP	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A
PH (211) FWoP	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A
PL (212) FWP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
PL (212) FWoP	5	5	5	5	5	20	20	10	10
SG (216) FWP	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A
SG (216) FWoP	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A

Table 40-6 Feedback Requirements With and Without PayPal (Continued)

Site (and ID)	Create A Store (Private Seller)	Create A Store (Business Seller)	List Fixed Price Item	Add Buy It Now Price To An Auction	Select 1-Day Duration	Specify Quantity >1 for Auction	Specify Quantity >1 for Fixed Price	Featured Plus	Home Page Featured
UK (3) FWP	5	0	5	5	5	10	10	10	10
UK (3) FWoP	10	0	10	10 or Direct Debit on file	10	20	20	10	10
US (0) FWP	0	0	0	5	5	15	15	10	10
US (0) FWoP	20	20	0	10 or ID verified	10	30	30	10	10

Miscellaneous Item Field Differences

[MotorsGermanySearchable](#) is currently only valid for DE (77).

To confirm that Home Page Featured is available for your site, call [GetCategoryFeatures](#) with the [SiteID](#) for your site and a [FeatureID](#) of [HomePageFeaturedEnabled](#). [HomePageFeaturedEnabled](#) is returned as an empty element if available for that site, and not returned the feature is unavailable.

Table 40-7 Miscellaneous Differences

Site (and ID)	BuyerProtection	Charity	Escrow	Get It Fast	Home Page Featured
AT (16)	yes	no	yes	yes	no
AU (15)	yes	no	yes	yes	yes
BEFR (23)	yes	no	yes	yes	yes
BENL (123)	yes	no	yes	yes	yes
CA (2)	yes	no	no	no	no
CAF (210)	yes	no	no	no	no
CH (193)	yes	no	no	yes	no
DE (77)	yes	no	yes	yes	yes
ES (186)	yes	no	yes	no	yes
FR (71)	yes	no	yes	yes	yes

Table 40-7 Miscellaneous Differences (Continued)

Site (and ID)	BuyerProtection	Charity	Escrow	Get It Fast	Home Page Featured
HK (201)	no	no	no	no	yes
IE (205)	yes	no	yes	yes	yes
IN (203)	no	no	no	no	yes
IT (101)	yes	no	yes	yes	yes
MY (207)	no	no	no	no	yes
NL (146)	yes	no	yes	yes	yes
PH (211)	no	no	no	no	yes
PL (212)	no	no	no	yes	no
SG (217)	no	no	no	no	yes
UK (3)	yes	yes	yes	yes	yes
US (0)	yes	yes	no	yes	yes

Payment Method Differences (PaymentMethod)

This section contains information about the [PaisaPay](#) payment method.

For information about specifying a payment method for a listing, including use of [GetCategoryFeatures](#) for determining the payment methods allowed for a category on a site, see “[Specifying a Payment Method](#)” on page 184.

The [PaisaPay](#) payment method is the online payment processing service for eBay India (IN), where buyers can pay sellers through a credit card or online bank transfer. The [PaisaPay](#) payment method is the required payment method for all categories on the IN site.

Sellers must register to use [PaisaPay](#) and can offer other payment methods after completing 20 lifetime listings. Listings that offer other payment methods still must offer [PaisaPay](#). The following example illustrates the use of [AddItem](#) offering [PaisaPay](#) and another payment method, as the seller has completed 20 lifetime listings:

Example 40-1 Specifying PaisaPay as a Payment Method

```
...  
<Item>  
...  
<PaymentMethods>PaisaPayAccepted</PaymentMethods>  
<PaymentMethods>COD</PaymentMethods>  
...  
</Item>  
...
```

PaisaPayEscrow on Selected IN Categories

The **PaisaPayEscrow** payment method adds three post-transaction event timelines, or escrow workflows, for both buyers and sellers to the regular **PaisaPay** payment method on selected categories on the IN site. An escrow workflow applies a timeline to events that buyers and sellers must complete after the buyer successfully submits a payment. The three escrow workflows available for **PaisaPayEscrow**, the **Default**, **WorkflowA**, and **WorkflowB** versions, each define different timelines for buyers and sellers for the post-transaction escrow events, such as the number of days a buyer has to complete a delivery confirmation. **PaisaPayEscrow** ensures that the buyer's funds are secure and are remitted to the seller only after the item has been shipped by the seller and received by the buyer. Sellers who become eligible for Express Remittance when using **PaisaPayEscrow** can also receive funds as soon as they enter shipping details for the transaction. For more information on qualifying for Express Remittance with **PaisaPayEscrow**, see the following link:

Express Remittance with PaisaPayEscrow
http://pages.ebay.in/paisapay/pspx_faqs.html#f2

When an item is sold using **PaisaPayEscrow**, the escrow workflow that is active for that category when a payment is completed, in addition to the shipping location, determines the post-transaction event timeline applied to the transaction. Different categories that allow **PaisaPayEscrow** support one of three escrow workflows above, so as to define an event timeline better-suited to the listed item. Note that **PaisaPayEscrow** can only be offered after the seller has registered to use it (see “[Using PaisaPayEscrowEMI Instead of PaisaPayEscrow](#)” on page 603 for more information).

Use **GetCategoryFeatures** to determine if the IN category supports **PaisaPayEscrow** by specifying **PaisaPayFullEscrowEnabled** in the **FeatureID** field of the request. If the category supports **PaisaPayEscrow**, you can then use **GetCategoryFeatures** to obtain the active escrow workflow on the category by specifying **INEscrowWorkflowTimeline** in the **FeatureID** field of the request. The response will indicate **Default**, **WorkflowA**, or **WorkflowB** in the **Category** field. Use **GetCategoryFeatures** to determine if the escrow workflow for a category has changed, although this will be rare. For more information on the escrow workflows, as well as information on workflow version changes across categories, see the following links:

PaisaPay Timelines
<http://pages.ebay.in/help/sell/paisapay-timelines.html>

PaisaPay Shipping
<http://pages.ebay.in/help/sell/paisapay-shipping.html>

For more information, see “[GetCategoryFeatures](#)” on page 808.

Using PaisaPayEscrowEMI Instead of PaisaPayEscrow

If the **PaisaPayEscrow** payment method is supported on a category, an optional sub-payment method called **PaisaPayEscrowEMI** (equal monthly installment) is also automatically available for qualified sellers. **PaisaPayEscrowEMI** allows buyers to pay in equal monthly installments using ICICI bank credit cards. To offer **PaisaPayEscrowEMI**, a seller must set the **Item.PaymentMethods** field in the **AddItem** request as follows:

Example 40-2 Specifying PaisaPayEscrowEMI as a Payment Method

```
...
<Item>
  ...
  <PaymentMethods>PaisaPayAccepted</PaymentMethods>
  <PaymentMethods>PaisaPayEscrow</PaymentMethods>
  <PaymentMethods>PaisaPayEscrowEMI</PaymentMethods>
  <PaymentMethods>COD</PaymentMethods>
  ...
</Item>
...
```

Note that **PaisaPayAccepted** and **PaisaPayEscrow** still must be specified in the request when offering **PaisaPayEscrowEMI**. The seller still can offer other payment methods if qualified.

The category's active escrow workflow applies to the transaction when a payment is completed using **PaisaPayEscrowEMI**. Note that **PaisaPayEscrowEMI** can only be offered after the seller has qualified for it.

For more information on **PaisaPay**, **PaisaPayEscrow** and **PaisaPayEscrowEMI**, including information on registering for **PaisaPay** and **PaisaPayEscrow**, see the PaisaPay FAQs.

PaisaPay FAQs

http://pages.ebay.in/paisapay/psx_faqs.html

Postal and Zip Codes (PostalCode)

This table lists the postal code formats for various sites that support search by proximity. If the country has N/A, that country does not support search by proximity.

Table 40-8 Postal Codes

Site (and ID)	Postal Code Format
AT (16)	4-digit numeric string
AU (15)	4-digit numeric string
BEFR (23)	4-digit numeric string
BENL (123)	4-digit numeric string
CA (2)	6-digit alphanumeric string

Table 40-8 Postal Codes (Continued)

Site (and ID)	Postal Code Format
CAF (210)	6-digit alphanumeric string
CH (193)	4-digit numeric string
DE (77)	5-digit numeric string
ES (186)	5-digit numeric string
FR (71)	5-digit numeric string
HK (201)	N/A
IE (205)	N/A
IN (203)	6-digit numeric string
IT (101)	5-digit numeric string
MY (207)	5-digit numeric string
NL (146)	4-digit number and 2 letters (with or without a space between the number and letters)
PH (211)	4-digit numeric string
PL (212)	5-digit numeric string
SG (216)	6-digit numeric string
UK (3)	5- to 8-digit alphanumeric string, w/space
US (0)	5-digit numeric string

Real Estate Ad Format Differences

Whether an AdType-type listing is supported for a given country.

Table 40-9 Real Estate Format Differences

Site (and ID)	AdType
AT (16)	no
AU (15)	no
BEFR (23)	no
BENL (123)	no
CA (2)	no
CAF (210)	no
CH (193)	no
DE (77)	yes
ES (186)	no
FR (71)	no
HK (201)	no
IE (205)	no

Table 40-9 Real Estate Format Differences (Continued)

Site (and ID)	AdType
IN (203)	yes
IT (101)	no
MY (207)	no
NL (146)	no
PH (211)	no
PL (212)	no
SG (216)	no
UK (3)	no
US (0)	yes

Search Type Differences

The tables below highlight the differences in search-related input fields for applicable international sites. A "Y" indicates that a particular search argument is available for a site.

Fields not shown in the matrix are equally supported across all eBay sites.

Table 40-10 Search by Title

Site (and ID)	Region	Currency	PayPal
AT (16)	N	N	N
AU (15)	Y	Y	N
BEFR (23)	N	Y	N
BENL (123)	N	Y	N
CA (2)	Y	Y	Y
DE (77)	Y	N	N
FR (71)	Y	Y	N
IE (205)	N	Y	Y
IN (203)	N	N	N
IT (101)	Y	Y	N
NL (146)	N	Y	N
PL (212)	Y	Y	N
UK (3)	Y	Y	Y
US (0)	Y	N	Y

Table 40-11 Search by Region or City

Site (and ID)	Query	CategoryID	Currency	SearchLocationFilter	SearchType	Order
AT (16)	N	N	N	N	N	N
AU (15)	Y	Y	Y	Y	Y	Y
BEFR (23)	N	N	N	N	N	N
BENL (123)	N	N	N	N	N	N
CA (2)	Y	Y	Y	Y	Y	Y
DE (77)	Y	Y	Y	Y	Y	Y
FR (71)	Y	Y	Y	Y	Y	Y
IE (205)	Y	Y	Y	N	Y	Y
IT (101)	Y	Y	Y	Y	Y	Y
NL (146)	N	N	N	N	N	N
PL (212)	Y	Y	Y	Y	Y	Y
UK (3)	Y	Y	Y	Y	Y	Y
US (0)	N	N	N	N	N	N

Table 40-12 Store Search

Site (and ID)	StoreSearchFilter	SearchFlags (SearchInDescription)
AT (16)	Y	Y
AU (15)	Y	Y
BEFR (23)	Y	Y
BENL (123)	Y	Y
CA (2)	Y	Y
DE (77)	Y	Y

Table 40-12 Store Search (Continued)

Site (and ID)	StoreSearchFilter	SearchFlags (SearchInDescription)
FR (71)	Y	Y
IE (205)	Y	Y
IN (203)	Y	Y
IT (101)	Y	Y
NL (146)	Y	Y
PL (212)	Y	Y
UK (3)	Y	Y
US (0)	Y	Y

Seller Level Differences (SellerLevel)

Table 40-13 Seller Level Differences

Site (and ID)	None	Bronze	Silver	Gold	Platinum	Titanium
AT (16)	yes	yes	yes	yes	yes	yes
AU (15)	yes	yes	yes	yes	yes	yes
BEFR (23)	yes	no	no	no	no	no
BENL (123)	yes	no	no	no	no	no
CA (2)	yes	yes	yes	yes	yes	yes
CAFR (210)	yes	yes	yes	yes	yes	yes
CH (193)	yes	yes	yes	yes	yes	yes
DE (77)	yes	yes	yes	yes	yes	yes
ES (186)	yes	no	no	no	no	no
FR (71)	yes	no	no	no	no	no
HK (201)	no	no	no	no	no	no

Table 40-13 Seller Level Differences (Continued)

Site (and ID)	None	Bronze	Silver	Gold	Platinum	Titanium
IE (205)	yes	no	no	yes	yes	yes
IT (101)	yes	no	no	no	no	no
MY (207)	no	no	no	no	no	no
NL (146)	yes	no	no	no	no	no
PH (211)	no	no	no	no	no	no
PL (212)	yes	yes	yes	yes	yes	yes
SG (216)	no	no	no	no	no	no
UK (3)	yes	no	no	yes	yes	yes
US (0)	yes	yes	yes	yes	yes	yes

Seller Payment Methods (SellerPaymentMethod)

Table 40-14 SellerPaymentMethods that Apply per Site

Site (and ID)	CreditCard	eBayDirectPay	DirectDebit	DirectDebitPendingSignatureMandate	NothingOnFile
AT (16)	yes		yes		yes
AU (15)	yes		yes		yes
BEFR (23)	yes				yes
BENL (123)	yes				yes
CA (2)	yes		yes		yes
CAF'R (210)	yes		yes		yes
CH (193)	yes				yes
DE (77)	yes		yes		yes
ES (186)					yes
FR (71)	yes		yes	yes	yes

Table 40-14 SellerPaymentMethods that Apply per Site (Continued)

Site (and ID)	CreditCard	eBayDirectPay	DirectDebit	DirectDebitPendingSignatureMandate	NothingOnFile
HK (201)	yes			yes	
IE (205)	yes	yes	yes	yes	yes
IN (203)					yes
IT (101)	yes	yes	yes	yes	yes
MY (207)	yes				yes
NL (146)	yes	yes	yes	yes	yes
PH (211)	yes				yes
PL (212)	yes	yes			yes
SG (216)	yes				yes
UK (3)	yes	yes	yes	yes	yes
US (0)	yes*	yes			yes

* Payment by debit card also supported

Shipping Region and Shipping Option Differences

ShippingOption was deprecated with compatibility level 383. Use ShipToLocation.

Table 40-15 Shipping Region and Option Differences

Site (and ID)	Africa	Asia	Caribbean	Europe	LatinAmerica	MiddleEast	NorthAmerica	Oceania	SouthAmerica	ShippingOption
AT (16)	no	no	no	yes	no	no	no	no	no	see note
AU (15)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes

Table 40-15 Shipping Region and Option Differences (Continued)

Site (and ID)	Africa	Asia	Caribbean	Europe	LatinAmerica	MiddleEast	NorthAmerica	Oceania	SouthAmerica	Shipping Option
BEFR (23)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
BENL (123)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
CA (2)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
CH (193)	no	no	no	yes	no	no	no	no	no	see note
DE (77)	no	no	no	yes	no	no	no	no	no	see note
ES (186)	no	no	no	yes	no	no	no	no	no	see note
FR (71)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
IT (101)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
NL (146)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
UK (3)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
US (0)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes

Note: If SitePlusRegions is specified, only Europe can be selected for other regions.

Promotional Sale Differences

The following table outlines the promotional sale logic and behavior differences by site.

Table 40-16 Site Differences for Promotional Sales

Site	Display Days left in Sale	Maximum Duration of Sale (in days)	Minimum Duration of Sale (in days)	Maximum % off Items	Minimum % off items	Minimum Time between sales (in days)	Minimum time Listed before item can go on Sale	Minimum Time at current Price before Item can go on sale	Edit Sale Allowed when active (Y/N)	Listing/ Editing Limit per day Basic Subscription	Listing/ Editing Limit per day Featured Subscription	Listing/ Editing Limit per day Anchor Subscription
AT (16)	14	14	3	50	5	7	7	7	N	50	250	2500
AU (15)	10	45	1	75	5	7	7	7	N	5000	5000	5000
CH (193)	14	14	3	50	5	7	7	7	N	50	250	2500

Table 40-16 Site Differences for Promotional Sales

Site	Display Days left in Sale	Maximum Duration of Sale (in days)	Minimum Duration of Sale (in days)	Maximum % of Items	Minimum % off items	Minimum Time between sales (in days)	Minimum time Listed before item can go on Sale	Minimum Time at current Price before Item can go on sale	Edit Sale Allowed when active (Y/N)	Listing/Editing Limit per day Basic Subscription	Listing/Editing Limit per day Featured Subscription	Listing/Editing Limit per day Anchor Subscription
DE (77)	14	14	3	50	5	7	7	7	N	50	250	2500
ES (186)	7	14	1	50	5	7	0	0	N	50	80	110
HK (201)	14	45	1	75	5	1	0	0	Y	250	2500	5000
IN (203)	7	30	1	75	5	1	0	0	Y	5000	5000	5000
SG (216)	7	45	1	75	5	1	0	0	N	250	2500	5000
UK (3)/IE (205)	14	14	1	50	5	14	28	28	N	250	2500	5000
US (0)	7	45	1	75	5	1	1	1	Y	250	2500	5000

Site IDs

- “Site IDs, Sorted by Number” on page 611
- “Site IDs, Sorted by Name” on page 612

For each eBay site, the table below lists the eBay site abbreviation (e.g., UK) and the corresponding site ID (e.g., 3), as well as the name of the site in English. You use the IDs in each call’s request URL (see “[Routing the Request \(Gateway URLs\)](#)” on page 87 for information about the request URL).

When setting object properties like `Item.Site`, you need to use site codes (`SiteCodeType`) instead. See the Schema Documentation for the current list of site codes. See “[Schema Reference Documentation](#)” on page 967 for information about where to find the Schema Documentation.

Use `GeteBayDetails`. with `DetailName` set to `SiteDetails` to retrieve details about specific eBay sites.

The annotations in `SiteCodeType` also provide useful information for eBay sites, such as the supported currency.

Table 40-17 Site IDs, Sorted by Number

ID (and abbreviation)	Name
US (0)	United States
CA (2)	Canada

Table 40-17 Site IDs, Sorted by Number (Continued)

ID (and abbreviation)	Name
UK (3)	United Kingdom
AU (15)	Australia
AT (16)	Austria
BEFR (23)	Belgium (French)
FR (71)	France
DE (77)	Germany
Motors (100)	US eBay Motors
IT (101)	Italy
BENL (123)	Belgium (Dutch)
NL (146)	Netherlands
ES (186)	Spain
CH (193)	Switzerland
HK (201)	Hong Kong
IN (203)	India
IE (205)	Ireland
MY (207)	Malaysia
CAF'R (210)	Canada (French)
PH (211)	Philippines
PL (212)	Poland
SG (216)	Singapore

Table 40-18 Site IDs, Sorted by Name

Abbreviation (and ID)	Name
AU (15)	Australia
AT (16)	Austria
BENL (123)	Belgium (Dutch)
BEFR (23)	Belgium (French)
CA (2)	Canada
CAF'R (210)	Canada (French)
FR (71)	France
DE (77)	Germany
HK (201)	Hong Kong
IE (205)	Ireland
IN (203)	India

Table 40-18 Site IDs, Sorted by Name (Continued)

Abbreviation (and ID)	Name
IT (101)	Italy
MY (207)	Malaysia
NL (146)	Netherlands
PH (211)	Philippines
PL (212)	Poland
SG (216)	Singapore
ES (186)	Spain
CH (193)	Switzerland
UK (3)	United Kingdom
US (0)	United States
Motors (100)	US eBay Motors

Site ID Filters for Search Calls (SiteIdFilter)

This table identifies which filters are valid for each site. The default filter is marked with an asterisk (*). This type is used in [GetCategoryListings](#) and [GetSearchResults](#) to enable a user to filter the category listings by site.

Table 40-19 Valid SiteIdFilterCodeTypes per Site

Abbreviation (and ID)	ListedInCurrencyImplied	LocatedInCountryImplied	AvailableInCountryImplied	SiteImplied	BelgiumListing
AT (16)	yes	yes*			
AU (15)	yes*	yes			
BEFR (23)	yes	yes	yes*		
BENL (123)	yes	yes	yes*		
CA (2)	yes	yes*			
CAF'R (210)	yes	yes*			
CH (193)	yes	yes*			
DE (77)	yes	yes*			
ES (186)	yes	yes*			
FR (71)	yes	yes*			

Table 40-19 Valid SiteIdFilterCodeTypes per Site (Continued)

Abbreviation (and ID)	ListedInCurrencyImplied	LocatedInCountryImplied	AvailableInCountryImplied	SiteImplied	BelgiumListing
HK (201)	yes	yes*		yes	
IE (205)		yes*	yes		
IT (101)			yes	yes*	
MY (207)	yes	yes*		yes	
NL (146)			yes	yes*	
PH (211)	yes	yes*		yes	
PL (212)		yes*	yes		
SG (216)	yes	yes*		yes	
UK (3)		yes*	yes		
US (0)			yes*		

Tax Field Differences (SalesTax, UseTaxTable)

For descriptions of shipping-related elements, see “[Shipping](#)” on page 312. For tax jurisdictions, see also “[Tax Jurisdictions](#)” on page 615.

Table 40-20 Tax Field Differences

Site (and ID)	SalesTaxPercent	SalesTaxState	ShippingIncludedInTax	Tax tables supported
AT (16)	no	no	no	no
AU (15)	no	no	no	no
BEFR (23)	no	no	no	no
BENL (123)	no	no	no	no
CA (2)	yes	yes	no	yes

Table 40-20 Tax Field Differences (Continued)

Site (and ID)	SalesTaxPercent	SalesTaxState	ShippingIncludedInTax	Tax tables supported
CAFR (210)	yes	yes	no	yes
CH (193)	no	no	no	no
DE (77)	no	no	no	no
ES (186)	no	no	no	no
FR (71)	no	no	no	no
HK (201)	no	no	no	no
IE (205)	yes	no	no	no
IN (203)	yes	yes	yes	no
IT (101)	no	no	no	no
MY (207)	no	no	no	no
NL (146)	no	no	no	no
PH (211)	no	no	no	no
PL (212)	no	no	no	no
SG (216)	no	no	no	no
UK (3)	yes	no	no	no
US (0)	yes	yes	yes	yes

Tax Jurisdictions

To determine whether tax tables are supported for a particular site, see “[Tax Field Differences \(SalesTax, UseTaxTable\)](#)” on page 614.

See also “[Enabling Multi-jurisdiction Sales Tax](#)” on page 231.

“[CA \(2\)](#)” on page 616

“[IN \(2\)](#)” on page 616

“[US \(0\)](#)” on page 618

Sites for which There are Currently no Tax Jurisdictions

AT, AT, AU, BEFR, BENL, CH, DE, ES, FR, HK, IE, IT, MY, NL, PH, SG, UK

CA (2)**Table 40-21 Jurisdictions, Canada**

Jurisdiction	TaxJurisdiction.JurisdictionID
Alberta	AB
British Columbia	BC
Manitoba	MB
New Brunswick	NB
Newfoundland and Labrador	NL
Nova Scotia	NS
Northwest Territories	NT
Nunavut	NU
Ontario	ON
Prince Edward Island	PE
Quebec	QC
Saskatchewan	SK
Yukon	YT

IN (2)**Table 40-22 Jurisdictions, India**

Jurisdiction	TaxJurisdiction.JurisdictionID
Andaman and Nicobar Islands	AN
Andhra Pradesh	AP
Arunachal Pradesh	AR
Assam	AS
Bihar	BR
Chandigarh	CH
Chhattisgarh	CT
Dadra and Nagar Haveli	DN
Daman and Diu	DD
Delhi	DL
Goa	GA
Gujarat	GJ
Haryana	HR
Himachal Pradesh	HP
Jammu and Kashmir	JK
Jharkhand	JH
Karnataka	KA

Table 40-22 Jurisdictions, India (Continued)

Jurisdiction	TaxJurisdiction.JurisdictionID
Kerala	KL
Lakshadweep	LD
Madhya Pradesh	MP
Maharashtra	MH
Manipur	MN
Meghalaya	ML
Mizoram	MZ
Nagaland	NL
Orissa	OR
Pondicherry	PY
Punjab	PB
Rajasthan	RJ
Sikkim	SK
Tamil Nadu	TN
Tripura	TR
Uttaranchal	UL
Uttar Pradesh	UP
West Bengal	WB

US (O)**Table 40-23 Jurisdictions, United States**

Jurisdiction	TaxJurisdiction.JurisdictionID
Alabama	AL
Alaska	AK
Arizona	AZ
Arkansas	AR
California	CA
Colorado	CO
Connecticut	CT
Delaware	DE
District of Columbia	DC
Florida	FL
Georgia	GA
Hawaii	HI
Idaho	ID
Illinois	IL
Indiana	IN
Iowa	IA
Kansas	KS
Kentucky	KY
Louisiana	LA
Maine	ME
Maryland	MD
Massachusetts	MA
Michigan	MI
Minnesota	MN
Mississippi	MS
Missouri	MO
Montana	MT
Nebraska	NE
Nevada	NV
New Hampshire	NH
New Jersey	NJ
New Mexico	NM
New York	NY
North Carolina	NC

Table 40-23 Jurisdictions, United States (Continued)

Jurisdiction	TaxJurisdiction.JurisdictionID
North Dakota	ND
Ohio	OH
Oklahoma	OK
Oregon	OR
Pennsylvania	PA
Rhode Island	RI
South Carolina	SC
South Dakota	SD
Tennessee	TN
Texas	TX
Utah	UT
Vermont	VT
Virginia	VA
Washington	WA
West Virginia	WV
Wisconsin	WI
Wyoming	WY

Part X

Platform Notifications Guide

Working with Platform Notifications	623
Wireless Device Applications	634
AskSellerQuestion	636
AuctionCheckoutComplete	638
BestOffer	642
BestOfferDeclined	644
BestOfferPlaced	646
BidPlaced	648
BidReceived	651
BulkDataExchangeJobCompleted	656
BuyerResponseDispute	751
Checkout	658
CheckoutBuyerRequestsTotal	665
CounterOfferReceived	663
EBAppealedCase	668

EBPOnHoldCase	669
EBPClosedAppeal	669
EBPClosedCase	669
EBPEscalatedCase	668
EBPMyPaymentDone	668
EBPMyPaymentDue	668
EBPMyResponseDue	667
EBPOtherPartyResponseDue	668
Feedback	675
FeedbackLeft	677
FeedbackReceived	679
FeedbackStarChanged	681
FixedPriceTransaction	684
INRBuyerOpenedDispute	685
INRBuyerRespondedToDispute	685
INRBuyerClosedDispute	686
INRSellerRespondedToDispute	686
ItemAddedToBidGroup	688
ItemAddedToWatchList	691
ItemClosed	694
ItemListed	699
ItemLost	703
ItemRemovedFromBidGroup	715
ItemRemovedFromWatchList	718
ItemRevised	722
ItemSold	725
ItemUnsold	729
ItemWon	733
MyMessagesAlert	738
MyMessagesAlertHeader	738
MyMessageseBayMessage	739

MyMessageseBayMessageHeader	739
MyMessagesM2MMessage	740
MyMessagesM2MMessageHeader	740
OutBid	742
ReadyToShip	758
ReadyForPayout	759
RefundFailure	759
RefundSuccess	759
ReturnClosed	748
ReturnCreated	747
ReturnDelivered	748
ReturnEscalated	749
ReturnRefundOverdue	748
ReturnSellerInfoOverdue	748
ReturnShipped	748
ReturnWaitingForSellerInfo	748
SecondChanceOffer	750
SellerOpenedDispute	756
SellerClosedDispute	754
SellerRespondedToDispute	760
TokenRevocation	762
UnmatchedPaymentReceived	759
WatchedItemEndingSoon	763

Chapter 41

Working with Platform Notifications

Platform notifications are triggered by events on the eBay site, such as the ending of a listing or the creation of a transaction. Notifications are SOAP messages about events on the eBay site, sent to subscribing applications.

You subscribe to notifications by setting your application's platform notification preferences, which tell eBay which events you want notification of and the URLs or email addresses to which you want the notifications delivered. Once you're subscribed, eBay Platform Notifications asynchronously pushes the notifications to your delivery location.

Subscribing to notifications is a good way to reduce the number of times your application has to execute transaction retrieval, feedback, and other routine API calls. Notifications do not count as API calls, and do not count against your daily API call limit.

You should use API calls to confirm the data received using notifications. For example, if you have subscribed to `AuctionCheckoutComplete` notification, ensure that you have set up periodic polling of `GetOrders`.

Your application should respond to notifications with a standard HTTP status of `200 OK`. Absent this response to a notification, eBay will attempt redelivery, but after a significant number of unacknowledged notifications, eBay may stop sending notifications to your application.

Platform notifications are not identical to the email messages that buyers and sellers might get when an item is listed, bid on, or purchased, though they may contain some of the same data.

For a brief demonstration of a user subscribing to and receiving notifications, see the API flow tutorial "Getting Notifications":

http://developer.ebay.com/DevZone/XML/docs/HowTo/Notifications/Notifications_listing.html

In this chapter:

- ["Notifications for Buyers and Sellers" on page 624](#)
- ["Subscribing to Platform Notifications" on page 624](#)
- ["Receiving Platform Notifications" on page 625](#)
- ["Informational Alerts" on page 631](#)

Notifications for Buyers and Sellers

Some notifications provide information that is specifically useful to sellers only, and these are restricted to sellers. Other notifications others provide information that is useful to buyers only, or to both buyers and sellers.

See <http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/types/NotificationEventTypeCodeType.html> for a description of each notification option and to whom it applies.

Subscribing to Platform Notifications

Your application can subscribe to notifications using [SetNotificationPreferences](#) (see [SetNotificationPreferences](#) in the Call Reference). See [NotificationEventTypeCodeType](#) for the kinds of events to which an application can subscribe.

You can check existing preferences with [GetNotificationPreferences](#) (see ["GetNotificationPreferences"](#) on page 843 for more details).

Your platform notification usage can be monitored with [GetNotificationsUsage](#). For more information on [GetNotificationsUsage](#) and how to use it to troubleshoot notification problems, see [GetNotificationsUsage](#) in the Call Reference.

Basic Requirements

The Compatible Application Check is not a requirement for using Platform Notifications.

Your application needs to be able to receive SOAP messages at a notification delivery URL or email address of your choice. You can then parse the SOAP message for the XML content contained in the SOAP envelope.

For more information about SOAP messages, see:

<http://www.w3.org/TR/SOAP/>

Notification Delivery URL Requirements

Notifications are delivered by an HTTP post to URLs or by a SMTP message to email addresses that you specify in your notification preferences.

A valid notification delivery URL begins with [http://](#) or [https://](#) and is well formed. Use [ApplicationDeliveryPreferences.ApplicationURL](#) to specify a URL to receive notifications for your application. Notifications will not be sent to any URL(s) if [ApplicationDeliveryPreferencesType.ApplicationEnable](#) is set to [Disable](#).

For more information about valid URLs, see:

URL Schemas (see [httpaddress](#))
W3C
http://www.w3.org/Addressing/URL/5_BNF.html

Notification delivery URLs can include query string variables. For example, the following URLs are valid:

http://my_hostname.com/cgi/my_notifications.asp
https://my_hostname.com/cgi/my_notifications.asp?myvar1=45&myvar2=123456789

Note: We recommend that you specify URLs that are currently functional. That is, your application should be ready to receive and respond to SOAP messages that are sent to these URLs. The subscription request will succeed if the delivery URLs are not functional URLs at the time the subscription request is made. However, if eBay begins sending notifications to the URLs and your application does not respond with a standard HTTP status 200 OK, eBay will interpret the lack of response as a notification failure. If a significant number of consecutive notifications fail, eBay may stop sending notifications for events associated with your AppId. If this occurs, please make sure the URLs are functional and then contact Developer Technical Support to reinstate your notification delivery.

Specifying Multiple Notification URLs

You can specify up to 25 notification URLs for an application. You define settings for up to 25 notification URLs (including the URL name in [DeliveryURLName](#)) in separate [ApplicationDeliveryPreferences.DeliveryURLDetails](#) containers in your [SetNotificationPreferences](#) request. You associate a user token with a notification URL (or up to 25 URLs) by using the token in a [SetNotificationPreferences](#) request that specifies the URL name in [SetNotificationPreferencesRequest.DeliveryURLName](#). For multiple URLs that have been defined in [ApplicationDeliveryPreferences.DeliveryURLDetails](#), use comma-separated format to enable them for a user token in [SetNotificationPreferencesRequest.DeliveryURLName](#).

Use [ApplicationDeliveryPreferences.ApplicationURL](#) to specify a default URL to receive notifications for your application. Notifications will not be sent to any URL(s) if [ApplicationDeliveryPreferencesType.ApplicationEnable](#) is set to [Disable](#).

Notifications for Fixed-Price Listings with Variations or Multiple Quantities

Because you can edit multiple-item, fixed-price listings, even after one or more of the items in that listing has been sold, sellers or buyers of items in a multiple-quantity listing can get a snapshot of the listing as it was at the time of a given purchase by invoking [GetItem](#) with a [Transaction ID](#) as input.

Sellers of an item are allowed to view all transactional snapshots of the listing, whereas buyers can only view the transactional snapshots for which they are the successful buyer. Past transactional views of an item are available for up to 90 days from the date the transaction took place.

You can set a transaction event to trigger your notifications, or you can set an event related to the entire listing to trigger your notifications. If you're using item transaction snapshot data, [Listing Status](#) may report an [Active](#) status because the listing was [Active](#) on the transaction date, but the listing may actually have [Ended](#).

Receiving Platform Notifications

After notification delivery is enabled for your application, eBay will send notifications related to listings that your application has submitted using [AddItem](#), [RelistItem](#), and [Reviseltem](#). When your application uses one of these calls, eBay identifies the application by its [AppId](#) (via the user token).

The application's [AppId](#) is then associated with the listing. This is how eBay knows which application to notify when transactional or feedback events occur in relation to the listing.

Also see "[Specifying Multiple Notification URLs](#)" on page 625.

Responding to Notifications

When a notification is sent, your application needs to respond with a standard HTTP status [200 OK](#). If you do not respond, eBay records the notification attempt as a failure, and the system does not resend the message. However, the database record for the event will still exist in the eBay system, so you can retrieve the event information by using the transaction-retrieval and feedback API functions (for example, [GetSellerEvents](#)).

If a significant number of notification failures occur consecutively (usually due to a problem with your delivery URLs or your application' ability to respond), eBay may stop sending notifications for events associated with your application's AppId.

SOAP Message Fields

Notifications are sent as SOAP messages. Each SOAP message consists of:

- An HTTP header.
- An XML processing instruction (the Platform Notifications feature only supports UTF-8 character encoding).
- A SOAP envelope containing a SOAP header and SOAP body.

The SOAP header contains the notification signature that is used for security, and the SOAP body contains the actual event.

HTTP Header

Each SOAP message contains an HTTP header that consists of the following pieces of information:

- **Post** -- contains the path of the URL that you provide.
- **PROTOCOL** -- HTTP/1.1 (The HTTP post is sent via HTTP, not HTTPS, because the data does not need the security of a certificate.)
- **HOST** -- contains the server specified in the URL that you provide
- **CONTENT_TYPE** -- contains "text/xml" and the XML encoding of the message
- **CONTENT_LENGTH** -- contains the length, in the form of a number, of the SOAP message
- **SOAPACTION** -- contains a URL that ends with the name of the event that the notification is being sent for. The value at the end will be one of the values specified in the [NotificationEventTypeCodeType](#) type in the schema.

This example shows an HTTP header for an **EndOfAuction** notification. This customer provided eBay with the URL `my_hostname.com/cgi/my_notifications.asp`.

Example 41-1 EndOfAuction Notification Header Example

```
POST: /cgi/my_notifications.asp
PROTOCOL: HTTP/1.1
HOST: my_hostname.com
CONTENT_TYPE: text/xml; charset="utf-8"
CONTENT_LENGTH: 1252
SOAPACTION: "HTTPS://developer.ebay.com/notification/EndOfAuction"
```

SOAP Message Header: Notification Signature

Each notification message contains a notification signature in the SOAP header for third-party application security. Your application should use the signature for message authentication to make sure the message was actually sent by eBay and that the notification signature was not copied by another party.

The notification signature is an MD5 hash signature that is generated using the following formula of concatenated strings, where DevId, AppId, and CertId are your application's Production Keys:

`eBayTime + DevId + AppId + CertId`

This hash is then Base64-encoded to reduce its length.

To have your application authenticate the signature:

- 1 Compute the MD5 hash using a standard routine, such as Microsoft crypto API, or OpenSSL open source.

You can find more information about these routines at:

<http://www.openssl.org/>

- 2 Convert the hash to Base64 encoding.
- 3 Compare the inputs in the notification signature to their actual values.

The `eBayTime` part of the signature should match the value of the **Timestamp** element in the SOAP message. Your application should check to make sure that the value of `eBayTime` is within 10

minutes of the actual time in GMT. The following code, written in C#, shows how to validate the notification hash:

Example 41-2 Example of How to Validate the Notification Hash (C#)

```
private bool CheckSignature(DateTime TimeStamp)
{
    const string AppId = "myproductionappid";
    const string DevId = "myproductiondevid";
    const string AuthCert = "myproductionauthcert";
    // Converts the TimeStamp back to universal time, because in .NET, XML schema time
    // values
    // are converted to local time
    // If you are retrieving the time stamp directly from the XML body of the notification
    // message, you would not need to convert it.
    string sig = TimeStamp.ToUniversalTime().ToString("yyyy-MM-ddTHH:mm:ss.ffffZ") + DevId +
        AppId + AuthCert;
    byte[] sigdata = System.Text.Encoding.ASCII.GetBytes(sig);
    System.Security.Cryptography.MD5 md5 = new
        System.Security.Cryptography.MD5CryptoServiceProvider();
    string md5Hash = System.Convert.ToBase64String(md5.ComputeHash(sigdata));
    return (mRequesterCredentials.NotificationSignature == md5Hash &&
        DateTime.Now.Subtract(TimeStamp).Duration().TotalMinutes <= 10);
}
```

SOAP Message Body

The following notification-related fields of **AbstractResponseType** are or can be returned in the body of a notification: **NotificationEventName**, **RecipientUserID**, **EIASToken**. (**NotificationSignature** is returned in the header.) See **AbstractResponseType** for descriptions of these fields.

Each SOAP message body contains a top-level element, the name of which is the call that was used to generate the data, with the word **Response** at the end. For example, the element name for the **EndOfAuction** call is **GetItemTransactionsResponse**. This element contains one attribute to specify the namespace: `xmlns="urn:ebay:apis:eBLBaseComponents"`. Child elements include some elements common to all notifications, such as **Timestamp**, which specifies the time the notification was sent. Other child elements are specific to the notification being sent. In the **EndOfAuction** notification, the **Item** element contains most of the data for the notification.

Notification Payloads

The content, or “payload,” of a notification is equivalent to the response from a corresponding API call.

With **SetNotificationPreferences**, you can use the **PayloadVersion** field to specify the API version for all notifications for the calling application. With **GetNotificationPreferences**, this field contains the API version for all notifications for the calling application.

See the following table ([Table 41-1](#)) for a list of notifications and their corresponding API calls.

Table 41-1 API Call Payloads for Notifications

Notification	API Call Payload
AskSellerQuestion	■ GetMemberMessages
AuctionCheckoutComplete	■ GetItemTransactions with default detail level
BestOffer	■ GetBestOffers
BestOfferDeclined	■ GetBestOffers with ReturnAll detail level
BestOfferPlaced	■ GetBestOffers with ReturnAll detail level
BidPlaced	■ GetItem with default detail level
BidReceived	■ GetItem with default detail level
BulkDataExchangeJobCompleted	■ See “ Data Fields Returned ” on page 656.
CheckoutBuyerRequestsTotal	■ GetItemTransactions with default detail level
Checkout	■ GetItemTransactions with default detail level
CounterOfferReceived	■ GetBestOffers with ReturnAll detail level
eBay Buyer Protection Notifications	■ See “ eBP Notification Structure ” on page 667.
EndOfAuction	■ GetItemTransactions with ReturnAll detail level
Feedback Notifications	■ GetFeedback with ReturnAll detail level
FeedbackLeft	■ GetFeedback with ReturnAll detail level
FeedbackReceived	■ GetFeedback with ReturnAll detail level
FeedbackStarChanged	■ GetFeedback
FixedPriceTransaction	■ GetItemTransactions with ReturnAll detail level
INR (ItemNotReceived) Notifications	■ GetDispute
ItemAddedToBidGroup	■ GetItem with default detail level
ItemAddedToWatchList	■ GetItem with default detail level
ItemClosed	■ GetItem with default detail level
ItemListed	■ GetItem with default detail level
ItemLost	■ GetItem with default detail level
ItemRemovedFromBidGroup	■ GetItem with default detail level
ItemRemovedFromWatchList	■ GetItem with default detail level
ItemRevised	■ GetItem with default detail level
ItemSold	■ GetItem
ItemUnsold	■ GetItem with default detail level
ItemWon	■ GetItem with default detail level
MyMessages Notification	■ GetMyMessages with ReturnHeaders or ReturnMessages detail level
OutBid	■ GetItem with default detail level
eBay Return Notifications	■ See “ Return Notification Structure ” on page 747.
SecondChanceOffer	■ GetItem with default detail level

Table 41-1 API Call Payloads for Notifications (Continued)

Notification	API Call Payload
BuyerResponseDispute	■ GetDispute
SellerClosedDispute	■ GetDispute
SellerOpenedDispute	■ GetDispute
Seller Payout Notifications	■ See “ Seller Payout Notifications ” on page 758.
SellerRespondedToDispute	■ GetDispute
TokenRevocation	■ GetTokenStatus
WatchedItemEndingSoon	■ GetItem

Receiving Notifications by Email

You can specify that you want to receive notifications by email, rather than posted to a server. To specify a default email URL for notifications, pass [ApplicationDeliveryPreferencesType](#) in [SetNotificationPreferences](#). In the [ApplicationDeliveryPreferencesType](#), set [ApplicationURL](#) to a URL that starts with `mailto://` or `mailto:` followed by a valid email address. For example, specify:

```
mailto://someone@xyz.com
mailto:anyone@acme.com
```

Use the same mechanism if you are specifying multiple locations for notifications (see “[Specifying Multiple Notification URLs](#)” on page 625).

Testing Platform Notifications

The Platform Notifications feature is supported in the Sandbox. Notifications are generated when certain events occur. To test the Platform Notifications feature in the Sandbox environment:

- Subscribe to the notifications you want (see “[Subscribing to Platform Notifications](#)” on page 624).
- Create listings, bids, disputes, or anything else you want to test, using the Sandbox web interface (at <http://sandbox.ebay.com>). Do something that generates a notification that your application is subscribed to, such as leaving feedback for a seller or having a buyer respond to a dispute.
- Make sure that your application can receive notifications.
- Make sure that your application responds to notifications with an HTTP status of [200 OK](#).
- Use [GetNotificationsUsage](#) to review the platform notification usage history for your application.
- Contact eBay Developer Technical Support if you have questions.

Informational Alerts

Account Activity Summary

A user can subscribe to receive an account activity summary via SMS. The user can specify the period (time range) for the account summary and can select how often the summary is to be sent.

This alert is not triggered by a direct event but rather by an eBay daemon process that monitors a subscription database.

Codelists, for your reference:

- Available summary periods: [SummaryWindowPeriodCodeType](#)
- Available summary frequencies: [SummaryFrequencyCodeType](#)

As noted in the following table, any period of activity can be returned on the user's choice of weekday. Periods that are 31 days or longer can also be returned via certain frequencies.

Table 41-2 Allowable Frequencies per Period

Window period	Frequency
All periods	<ul style="list-style-type: none">■ Weekly, on the weekday of the user's choosing (e.g. EveryTuesday).
31 days	<ul style="list-style-type: none">■ Every31Days
Current month	<ul style="list-style-type: none">■ On a selected date of every month (e.g. MonthlyOn4th).
Last month	<ul style="list-style-type: none">■ On the first day of every month (the summary covers the previous month's activity): MonthlyOn1st
Last 60 days	<ul style="list-style-type: none">■ Every60Days

Adding or Updating a Schedule

Use [SummarySchedule](#) in combination with [UserDeliveryPreferenceArray.NotificationEnable](#) to set [EventType](#) to [AccountSummary](#) and to set [EventEnable](#) to [Enable](#). Also set [UserData.SummarySchedule.EventType](#) to [AccountSummary](#), and set [SummaryPeriod](#) and [Frequency](#) as desired. To delete all schedules for a user, set [EventEnable](#) to [Disable](#), and omit [SummarySchedule](#) from the [UserData](#) container (or omit the [UserData](#) container entirely).

The following example creates a new schedule for the user. Every Monday, the user will receive a summary of the last 31 days' activities.

Example 41-3 Within SetNotificationPreferences: Creating/Adding

```
<UserDeliveryPreferenceArray>
  <NotificationEnable>
    <EventType>AccountSummary</EventType>
    <EventEnable>Enable</EventEnable>
  </NotificationEnable>
</UserDeliveryPreferenceArray>
<UserData>
  <SummarySchedule>
    <EventType>AccountSummary</EventType>
    <SummaryPeriod>Last31Days</SummaryPeriod>
    <Frequency>EveryMonday</Frequency>
  </SummarySchedule>
</UserData>
```

Deleting Schedules

To delete all schedules for a user, omit `UserData` from the call to `SetNotificationPreferences`:

Example 41-4 Within SetNotificationPreferences: Deleting All

```
<UserDeliveryPreferenceArray>
  <NotificationEnable>
    <EventType>AccountSummary</EventType>
    <EventEnable>Disable</EventEnable>
  </NotificationEnable>
</UserDeliveryPreferenceArray>
```

To delete a specific schedule, set `UserData.SummarySchedule.EventType` to `AccountSummary`, and set `SummaryPeriod` and `Frequency` to match the schedule to be deleted.

The following example deletes the “EveryMonday, Last31Days” schedule for the user:

Example 41-5 Within SetNotificationPreferences: Deleting Selected

```
<UserDeliveryPreferenceArray>
  <NotificationEnable>
    <EventType>AccountSummary</EventType>
    <EventEnable>Disable</EventEnable>
  </NotificationEnable>
</UserDeliveryPreferenceArray>
<UserData>
  <SummarySchedule>
    <EventType>AccountSummary</EventType>
    <SummaryPeriod>Last31Days</SummaryPeriod>
    <Frequency>EveryMonday</Frequency>
  </SummarySchedule>
</UserData>
```

To delete all schedules matching a particular frequency regardless of summary period, set `Frequency` to that particular frequency and omit `SummaryPeriod`.

The following example deletes all of the user's schedules for Monday:

Example 41-6 Within SetNotificationPreferences: Deleting Selected

```
<UserDeliveryPreferenceArray>
  <NotificationEnable>
    <EventType>AccountSummary</EventType>
    <EventEnable>Disable</EventEnable>
  </NotificationEnable>
</UserDeliveryPreferenceArray>
<UserData>
  <SummarySchedule>
    <EventType>AccountSummary</EventType>
    <Frequency>EveryMonday</Frequency>
  </SummarySchedule>
</UserData>
```

Similarly, you can delete all schedules matching a **SummaryPeriod** by providing **SummaryPeriod** and omitting **Frequency**.

Chapter 42

Wireless Device Applications

Note: Wireless-related API features you can find in the schema can only be used by parties selected and approved by eBay.

This chapter discusses API support for wireless devices.

["Support for Wireless Applications" on page 634](#)

["Notification Events for Wireless" on page 634](#)

Support for Wireless Applications

Note: Wireless-related API features you can find in the schema can only be used by parties selected and approved by eBay.

Notification Events for Wireless

Note: Wireless-related API features you can find in the schema can only be used by parties selected and approved by eBay.

Some notifications can be received via wireless device.

Table 42-1 Applicability of Notifications to Wireless (SMS)

Notification	Available for wireless
AskSellerQuestion	yes
AuctionCheckoutComplete	no
BestOffer	no
BestOfferDeclined	no
BestOfferPlaced	no
BidPlaced	no
BidReceived	no
CheckoutBuyerRequestsTotal	no
CounterOfferReceived	no
EndOfAuction	no
Feedback Notifications	no
FeedbackLeft	no
FeedbackReceived	no

Table 42-1 Applicability of Notifications to Wireless (SMS)

Notification	Available for wireless
FeedbackStarChanged	no
FixedPriceTransaction	no
INR (ItemNotReceived) Notifications	no
ItemAddedToBidGroup	no
ItemAddedToWatchList	no
ItemLost	no
ItemRemovedFromBidGroup	no
ItemRemovedFromWatchList	no
ItemSold	no
ItemUnsold	no
ItemWon	no
MyMessages Notification	no
OutBid	yes
SecondChanceOffer	yes
BuyerResponseDispute	no
SellerClosedDispute	no
SellerOpenedDispute	no
SellerRespondedToDispute	no
WatchedItemEndingSoon	yes

Additional Steps for WatchedItemEndingSoon Subscription

On the eBay web site notifications preferences page, a user can request to be notified about items ending soon and can set the amount of time prior to the end of the listing. A user can also subscribe via API.

In addition to the typical steps for setting notification preferences, there are a few more steps for subscribing a user for the [WatchedItemEndingSoon](#) notification:

- 1 Set [EventProperty.EventType](#) to [WatchedItemEndingSoon](#).
- 2 Set [EventProperty.Name](#) to [TimeLeft](#).
- 3 Set [EventProperty.Value](#) to the number of minutes before the end of the listing that the user is to be notified.

Chapter 43

AskSellerQuestion

eBay sends an **AskSellerQuestion** notification when a user asks a question about one of a seller's active listings.

Platform Choice

When a user requests, via the notification preferences web page, to be notified of this event, the default is for the notification to be sent via API ("Platform"). The user can choose instead to be notified via wireless ("SMS").

Data Fields Returned

The data fields returned are the same as those returned by [GetMemberMessages](#). Most of the useful information in the body of this notification is nested within the [MemberMessage.MemberMessageExchange](#) property.

Here are some of main properties:

Table 43-1

Property	Use
<code>Item.ItemID</code>	Item ID of the item the user asked a question about.
<code>Question.QuestionType</code>	Type of question being asked
<code>Question.SenderID</code>	User who sent the question
<code>Question.Body</code>	Text of the question being asked.
<code>CreationDate</code>	Time the question was created (the time it was sent).

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 43-1 Sample AskSellerQuestion Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" 
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance">
    <soapenv:Header>
        <ebl:RequesterCredentials soapenv:mustUnderstand="0"
            xmlns:ns="urn:ebay:apis:eBLBaseComponents" xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
            <ebl:NotificationSignature
                xmlns:ebl="urn:ebay:apis:eBLBaseComponents">AKurONibBT6zkNCsQ27rtg==
            </ebl:NotificationSignature>
        </ebl:RequesterCredentials>
    </soapenv:Header>
    <soapenv:Body>
        <GetMemberMessagesResponse xmlns="urn:ebay:apis:eBLBaseComponents">
            <Timestamp>2005-04-26T21:49:07.567Z</Timestamp>
            <Ack>Success</Ack>
            <CorrelationID>00000000-00000000-00000000-00000000-00000000-
                0000000000</CorrelationID>
            <Version>405</Version>
            <Build>20050419162249</Build>
            <NotificationEventName>AskSellerQuestion</NotificationEventName>
            <RecipientUserId>sampleseller</RecipientUserId>
            <MemberMessage>
                <MemberMessageExchange>
                    <Item>
                        <ItemID>4502794599</ItemID>
                        <ListingDetails>
                            <StartTime>2005-04-26T21:45:04.000Z</StartTime>
                            <EndTime>2005-05-01T21:45:04.000Z</EndTime>
                        </ListingDetails>
                        <SellingStatus>
                            <CurrentPrice currencyID="USD">0.0</CurrentPrice>
                        </SellingStatus>
                        <Title>Toy car - antique</Title>
                    </Item>
                    <Question>
                        <MessageType>AskSellerQuestion</MessageType>
                        <QuestionType>General</QuestionType>
                        <DisplayToPublic>false</DisplayToPublic>
                        <SenderID>samplebuyer</SenderID>
                        <RecipientID>397681</RecipientID>
                        <Body>Text of the question</Body>
                        <MessageID>695</MessageID>
                    </Question>
                    <MessageStatus>Unanswered</MessageStatus>
                    <CreationDate>2005-04-26T21:46:26.000Z</CreationDate>
                    <LastModifiedDate>2005-04-26T21:46:26.000Z</LastModifiedDate>
                </MemberMessageExchange>
            </MemberMessage>
            <PaginationResult>
                <TotalNumberOfPages>1</TotalNumberOfPages>
                <TotalNumberOfEntries>1</TotalNumberOfEntries>
            </PaginationResult>
            <HasMoreItems>false</HasMoreItems>
        </GetMemberMessagesResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Chapter 44

AuctionCheckoutComplete

An [AuctionCheckoutComplete](#) notification is sent when a buyer completes the checkout process for an auction item or for a fixed price item.

For checkouts that involve more than one item, [AuctionCheckoutComplete](#) is sent only for the first item. Because its payload is based on [GetItemTransactions](#), only a single item is included in the notification.

Usually, an [AuctionCheckoutComplete](#) notification occurs only once per auction. However, if the payment for checkout fails, the checkout status will change from “complete” to “incomplete.” If the buyer then successfully pays for the item via checkout, the checkout status will change back to “complete” again. When the checkout status changes back to “complete,” another [AuctionCheckoutComplete](#) notification will be sent.

Data Fields Returned

The data fields in the notification are the same as those returned by the [GetItemTransactions](#) call with the default detail level.

See [“SOAP Message Body”](#) on page 628 for details on the standard elements returned for all notifications.

Example

Example 44-1 Sample AuctionCheckoutComplete Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0"
xmlns:ns="urn:ebay:apis:eBLBaseComponents" xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature
  xmlns:ebl="urn:ebay:apis:eBLBaseComponents">AKurONibBT6zkNCsQ27rtg==
</ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetItemTransactionsResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2005-04-27T17:48:03.545Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>00000000-00000000-00000000-00000000-00000000-0000000000
</CorrelationID>
<Version>407</Version>
<Build>20050426141851</Build>
<NotificationEventName>AuctionCheckoutComplete</NotificationEventName>
<PaginationResult>
<TotalNumberOfPages>1</TotalNumberOfPages>
<TotalNumberOfEntries>1</TotalNumberOfEntries>
</PaginationResult>
<HasMoreTransactions>false</HasMoreTransactions>
<TransactionsPerPage>100</TransactionsPerPage>
<PageNumber>1</PageNumber>
<ReturnedTransactionCountActual>1</ReturnedTransactionCountActual>
<Item>
<AutoPay>false</AutoPay>
<BuyerProtection>ItemIneligible</BuyerProtection>
<Currency>USD</Currency>
<ItemID>2211777403</ItemID>
<ListingType>Chinese</ListingType>
<PaymentMethods>PaymentSeeDescription</PaymentMethods>
<PrivateListing>true</PrivateListing>
<Quantity>1</Quantity>
<Seller>
<AboutMePage>true</AboutMePage>
<EIASToken>Y+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wJnY+gD5iFoA+dj6x9nY+seQ==</EIASToken>
<Email>sampleseller@sampleseller.com</Email>
<FeedbackScore>4119</FeedbackScore>
<FeedbackRatingStar>Red</FeedbackRatingStar>
<IDVerified>true</IDVerified>
<NewUser>false</NewUser>
<RegistrationDate>2004-10-03T18:28:52.000Z</RegistrationDate>
<Site>US</Site>
<Status>Confirmed</Status>
<UserID>sampleseller</UserID>
<UserIDChanged>false</UserIDChanged>
<UserIDLastChanged>2001-01-19T17:21:01.000Z</UserIDLastChanged>

```

Example 44-1 Sample AuctionCheckoutComplete Notification

```
<SellerInfo>
  <AllowPaymentEdit>true</AllowPaymentEdit>
  <CheckoutEnabled>true</CheckoutEnabled>
  <CIPBankAccountStored>false</CIPBankAccountStored>
  <GoodStanding>true</GoodStanding>
  <MerchandizingPref>OptIn</MerchandizingPref>
  <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
  <SellerLevel>None</SellerLevel>
  <StoreOwner>true</StoreOwner>
  <StoreURL>http://www.ebaystores.ebay.com/id=178713</StoreURL>
</SellerInfo>
</Seller>
<SellingStatus>
  <ConvertedCurrentPrice currencyID="USD">100.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">100.0</CurrentPrice>
  <QuantitySold>1</QuantitySold>
</SellingStatus>
<Site>US</Site>
<Title>GetItem Test</Title>
</Item>
<TransactionArray>
  <Transaction>
    <AmountPaid currencyID="USD">100.0</AmountPaid>
    <AdjustmentAmount currencyID="USD">0.0</AdjustmentAmount>
    <ConvertedAdjustmentAmount currencyID="USD">0.0</ConvertedAdjustmentAmount>
    <Buyer>
      <AboutMePage>false</AboutMePage>
      <EIASToken>nY+sHZ2PrBndj6wVnY+sEZ2PrA2dj6wJnY+gD5iFogidj6x9nY+seQ==</EIASToken>
      <Email>samplebuyer@samplebuyer.com</Email>
      <FeedbackScore>58</FeedbackScore>
      <FeedbackRatingStar>Blue</FeedbackRatingStar>
      <IDVerified>true</IDVerified>
      <NewUser>false</NewUser>
      <RegistrationDate>1999-01-19T08:00:00.000Z</RegistrationDate>
      <Site>US</Site>
      <Status>Confirmed</Status>
      <UserID>samplebuyer</UserID>
      <UserIDChanged>false</UserIDChanged>
      <UserIDLastChanged>2001-01-19T18:42:14.000Z</UserIDLastChanged>
    <BuyerInfo>
      <ShippingAddress>
        <Name>new_Name</Name>
        <Street1>a new street1</Street1>
        <Street2>new street 2</Street2>
        <CityName>my new town</CityName>
        <StateOrProvince>CA</StateOrProvince>
        <Country>US</Country>
        <CountryName>United States</CountryName>
        <Phone>(111) 111-1</Phone>
        <PostalCode>95555</PostalCode>
        <AddressID>191925</AddressID>
        <AddressOwner>eBay</AddressOwner>
      </ShippingAddress>
    </BuyerInfo>
  </Buyer>
</Transaction>
</TransactionArray>
```

Example 44-1 Sample AuctionCheckoutComplete Notification

```
<ShippingDetails>
    <ChangePaymentInstructions>true</ChangePaymentInstructions>
    <InsuranceFee currencyID="USD">0.0</InsuranceFee>
    <InsuranceOption>NotOffered</InsuranceOption>
    <InsuranceWanted>false</InsuranceWanted>
    <PaymentEdited>false</PaymentEdited>
    <SalesTax>
        <SalesTaxPercent>0.0</SalesTaxPercent>
        <ShippingIncludedInTax>false</ShippingIncludedInTax>
    </SalesTax>
    <ShippingServiceOptions>
        <ShippingService>ShippingMethodStandard</ShippingService>
        <ShippingServiceCost currencyID="USD">0.0</ShippingServiceCost>
        <ShippingServiceAdditionalCost currencyID="USD">0.0
            </ShippingServiceAdditionalCost>
        <ShippingServicePriority>1</ShippingServicePriority>
    </ShippingServiceOptions>
</ShippingDetails>
<ConvertedAmountPaid currencyID="USD">100.0</ConvertedAmountPaid>
<ConvertedTransactionPrice currencyID="USD">100.0</ConvertedTransactionPrice>
<CreatedDate>2005-04-09T01:27:52.000Z</CreatedDate>
<DepositType>None</DepositType>
<QuantityPurchased>1</QuantityPurchased>
<Status>
    <eBayPaymentStatus>NoPaymentFailure</eBayPaymentStatus>
    <CheckoutStatus>CheckoutComplete</CheckoutStatus>
    <LastTimeModified>2005-04-27T17:46:04.000Z</LastTimeModified>
    <PaymentMethodUsed>PaymentSeeDescription</PaymentMethodUsed>
    <CompleteStatus>Complete</CompleteStatus>
</Status>
<TransactionID>0</TransactionID>
<TransactionPrice currencyID="USD">100.0</TransactionPrice>
<BestOfferSale>false</BestOfferSale>
<ShippingServiceSelected>
    <ShippingInsuranceCost currencyID="USD">0.0</ShippingInsuranceCost>
    <ShippingService>ShippingMethodStandard</ShippingService>
    <ShippingServiceCost currencyID="USD">0.0</ShippingServiceCost>
</ShippingServiceSelected>
</Transaction>
</TransactionArray>
<PayPalPreferred>true</PayPalPreferred>
</GetItemTransactionsResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 45

BestOffer

eBay sends a **BestOffer** message when a bidder makes an offer on a listing that is enabled for the Best Offer feature. The notification is not sent to the buyer or seller, but only to a third party such as an eBay development partner. The flow of events is this:

- A third party subscribes to the **BestOffer** notification on behalf of a seller who has opted a listing into the Best Offer feature.
- eBay sends a notification to the third party each time a bidder makes an offer on that listing.

Data Fields Returned

The data fields the **BestOffer** notification returns are the same as those returned by **GetBestOffersResponse**. Check **GetBestOffersResponse** in the generated WsdlDoc documentation.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

Example 45-1 Sample BestOffer Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Header>
        <ebl:RequesterCredentials soapenv:mustUnderstand="0">
            xmlns:ns="urn:ebay:apis:eBLBaseComponents"
            xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
                <ebl:NotificationSignature>abc123==</ebl:NotificationSignature>
            </ebl:RequesterCredentials>
        </soapenv:Header>
    <soapenv:Body>
        <GetBestOffersResponse xmlns="urn:ebay:apis:eBLBaseComponents">
            <Timestamp>2005-01-20T01:46:15.839Z</Timestamp>
            <Ack>Success</Ack>
            <CorrelationID>67187690</CorrelationID>
            <Version>361</Version>
            <Build>20050119095443</Build>
            <NotificationEventName>BestOffer</NotificationEventName>
            <BestOfferArray>
                <BestOffer>
                    <BestOfferID>4521</BestOfferID>
                    <ExpirationTime>2005-01-21T02:46:23.000Z</ExpirationTime>
                    <Buyer>
                        <Email>samplebuyer@abc123.com</Email>
                        <FeedbackScore>115</FeedbackScore>
                        <RegistrationDate>1999-01-19T08:00:00.000Z</RegistrationDate>
                        <UserID>samplebuyer</UserID>
                    </Buyer>
                    <Currency>USD</Currency>
                    <Price>0.5</Price>
                    <Status>Pending</Status>
                    <Quantity>1</Quantity>
                </BestOffer>
            </BestOfferArray>
            <Item>
                <BuyItNowPrice>1.0</BuyItNowPrice>
                <Currency>USD</Currency>
                <ListingDetails>
                    <EndTime>2005-01-24T02:45:36.000Z</EndTime>
                </ListingDetails>
                <Location>San Jose</Location>
                <Title>BestOffer notification test.</Title>
            </Item>
        </GetBestOffersResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Chapter 46

BestOfferDeclined

eBay sends a **BestOfferDeclined** notification to third parties subscribed on the buyer's behalf whenever a seller rejects the buyer's Best Offer. The typical flow of events is:

- A third party subscribes to the **BestOfferDeclined** notification on behalf of a buyer.
- The buyer's Best Offer is rejected by the seller.
- eBay sends the **BestOfferDeclined** notification to the third party, indicating the buyer's Best Offer was rejected.

eBay also sends **BestOfferDeclined** notifications to the third party each time a seller rejects a Best Offer placed by the buyer, either through the eBay sites or programmatically from an API or SDK.

Data Fields Returned

The data fields the **BestOfferDeclined** notification returns are the same as those returned by **GetBestOfferResponse** with the **ReturnAll** detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 46-1 Sample BestOfferDeclined Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">NSQ3JxY+kDkwTP+oFPT6g==</
ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetBestOffersResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-02-13T03:16:34.637Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>137540530</CorrelationID>
<Version>553</Version>
<Build>20050429131630</Build>
<NotificationEventName>BestOfferDeclined</NotificationEventName>
<RecipientUserID>Seller1</RecipientUserID>
<BestOfferArray>
<BestOffer>
<BestOfferID>6712</BestOfferID>
<ExpirationTime>2008-02-15T03:16:17.000Z</ExpirationTime>
<Buyer>
<Email>buyer@email.com</Email>
<FeedbackScore>1</FeedbackScore>
<RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
<UserID>Buyer1</UserID>
</Buyer>
<Price currencyID="USD">500.0</Price>
<Status>Declined</Status>
<Quantity>1</Quantity>
<BestOfferCodeType>BuyerBestOffer</BestOfferCodeType>
</BestOffer>
</BestOfferArray>
<Item>
<BuyItNowPrice currencyID="USD">1000.0</BuyItNowPrice>
<ItemID>250000627083</ItemID>
<ListingDetails>
<EndTime>2008-02-18T03:15:32.000Z</EndTime>
</ListingDetails>
<Location>San Jose, CA</Location>
<Title>Test Norman BestOfferDeclined notify</Title>
</Item>
</GetBestOffersResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 47

BestOfferPlaced

eBay sends a **BestOfferPlaced** notification to third parties subscribed on the buyer's behalf whenever a buyer places a Best Offer. A buyer can place a Best Offer by making an initial Best Offer or by countering a counter offer. The typical flow of events is:

- A third party subscribes to the **BestOfferPlaced** notification on behalf of a buyer.
- The buyer places a Best Offer or counters a counter offer.
- eBay sends the **BestOfferPlaced** notification to the third party, indicating the buyer placed a Best Offer.

eBay also sends BestOfferPlaced notifications to the third party each time a buyer places a Best Offer, either through the eBay sites or programmatically from an API or SDK.

Data Fields Returned

The data fields the **BestOfferPlaced** notification returns are the same as those returned by **GetBestOffersResponse** at the ReturnAll detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 47-1 Sample BestOfferPlaced Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">WSS9Y+tw+UfADY4rJF2YZQ==</
ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetBestOffersResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-02-13T02:38:17.950Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>137540360</CorrelationID>
<Version>553</Version>
<Build>e553_core_Bundled_6057805_R1</Build>
<NotificationEventName>BestOfferPlaced</NotificationEventName>
<RecipientUserID>...buyer userid goes here...</RecipientUserID>
<BestOfferArray>
<BestOffer>
<BestOfferID>6710</BestOfferID>
<ExpirationTime>2008-02-15T02:38:02.000Z</ExpirationTime>
<Buyer>
<Email>samplebuyer@abc123.com</Email>
<FeedbackScore>1</FeedbackScore>
<RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
<UserID>Buyer1</UserID>
</Buyer>
<Price currencyID="USD">1000.0</Price>
<Status>Pending</Status>
<Quantity>1</Quantity>
<BestOfferCodeType>BuyerBestOffer</BestOfferCodeType>
</BestOffer>
</BestOfferArray>
<Item>
<BuyItNowPrice currencyID="USD">1000.0</BuyItNowPrice>
<ItemID>250000627062</ItemID>
<ListingDetails>
<EndTime>2008-02-18T02:37:17.000Z</EndTime>
</ListingDetails>
<Location>San Jose, CA</Location>
<Title>Test BestOfferPlaced notify</Title>
</Item>
</GetBestOffersResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 48

BidPlaced

eBay sends a **BidPlaced** notification to third parties subscribed on the buyer's behalf whenever a buyer places a bid on an auction item. The typical flow of events is:

A third party subscribes to the **BidPlaced** notification on behalf of a buyer.

The buyer bids on an auction item.

eBay sends the **BidPlaced** notification to the third party, indicating that the buyer has bid on an auction item.

eBay also sends **BidPlaced** notifications to the third party for all auction items the buyer bids on, either through the eBay sites or programmatically from an API or SDK.

Data Fields Returned

The data fields the **BidPlaced** notification returns are the same as those returned by **GetItemResponse** with the default detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 48-1 Sample BidPlaced Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents" xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">fgJAxkils3LD7GF/IcSDmg==</ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-02-13T03:28:25.715Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>137540850</CorrelationID>
<Version>553</Version>
<Build>e553_core_Bundled_6057805_R1</Build>
<NotificationEventName>BidPlaced</NotificationEventName>
```

Example 48-1 Sample BidPlaced Notification

```

<RecipientUserID>Buyer1</RecipientUserID>
<Item>
    <AutoPay>false</AutoPay>
    <BuyerProtection>ItemEligible</BuyerProtection>
    <BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
    <Country>US</Country>
    <Currency>USD</Currency>
    <GiftIcon>0</GiftIcon>
    <HitCounter>NoHitCounter</HitCounter>
    <ItemID>250000627094</ItemID>
    <ListingDetails>
        <Adult>false</Adult>
        <BindingAuction>false</BindingAuction>
        <CheckoutEnabled>true</CheckoutEnabled>
        <ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
        <ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
        <HasReservePrice>false</HasReservePrice>
        <StartTime>2008-02-13T03:27:07.000Z</StartTime>
        <EndTime>2008-02-18T03:27:07.000Z</EndTime>
        <ViewItemURL>http://cgi.qa-api012.qa.ebay.com/ws/
eBayISAPI.dll?ViewItem&item=250000627094&category=1463</ViewItemURL>
        <HasUnansweredQuestions>false</HasUnansweredQuestions>
        <HasPublicMessages>false</HasPublicMessages>
        <ExpressListing>false</ExpressListing>
        <ViewItemURLForNaturalSearch>http://cgi.qa-api012.qa.ebay.com/Test-Bid-Placed-Received-
notify_WQQItemz250000627094QQcategoryZ1463QcmdZViewItem</ViewItemURLForNaturalSearch>
    </ListingDetails>
    <ListingDuration>Days_5</ListingDuration>
    <ListingType>Chinese</ListingType>
    <Location>San Jose, CA</Location>
    <PaymentMethods>PaymentSeeDescription</PaymentMethods>
    <PaymentMethods>PayPal</PaymentMethods>
    <PrimaryCategory>
        <CategoryID>1463</CategoryID>
        <CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
    </PrimaryCategory>
    <PrivateListing>false</PrivateListing>
    <Quantity>1</Quantity>
    <ReviseStatus>
        <ItemRevised>false</ItemRevised>
    </ReviseStatus>
    <Seller>
        <AboutMePage>false</AboutMePage>
        <Email>Invalid Request</Email>
        <FeedbackScore>100</FeedbackScore>
        <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
        <FeedbackPrivate>false</FeedbackPrivate>
        <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
        <IDVerified>false</IDVerified>
        <eBayGoodStanding>true</eBayGoodStanding>
        <NewUser>false</NewUser>
        <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
        <Site>US</Site>
        <Status>Confirmed</Status>
        <UserID>Seller1</UserID>
        <UserIDChanged>false</UserIDChanged>
        <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
        <VATStatus>NoVATTax</VATStatus>
        <SellerInfo>
            <AllowPaymentEdit>true</AllowPaymentEdit>
            <CheckoutEnabled>true</CheckoutEnabled>
            <CIPBankAccountStored>false</CIPBankAccountStored>
            <GoodStanding>true</GoodStanding>
            <MerchandizingPref>Opt In</MerchandizingPref>
            <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
            <SellerLevel>None</SellerLevel>
            <StoreOwner>false</StoreOwner>
            <ExpressEligible>false</ExpressEligible>
            <ExpressWallet>false</ExpressWallet>
            <SafePaymentExempt>true</SafePaymentExempt>
        </SellerInfo>
        <MotorsDealer>false</MotorsDealer>
    </Seller>

```

Example 48-1 Sample BidPlaced Notification

```
<SellingStatus>
  <BidCount>1</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.0</CurrentPrice>
  <HighBidder>
    <AboutMePage>false</AboutMePage>
    <EIASToken>nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wJkoejAJKFqQWdj6x9nY+seQ==</EIASToken>
    <Email>buyer1@email.com</Email>
    <FeedbackScore>1</FeedbackScore>
    <PositiveFeedbackPercent>66.7</PositiveFeedbackPercent>
    <FeedbackPrivate>false</FeedbackPrivate>
    <FeedbackRatingStar>None</FeedbackRatingStar>
    <IDVerified>false</IDVerified>
    <eBayGoodStanding>true</eBayGoodStanding>
    <NewUser>false</NewUser>
    <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
    <Site>US</Site>
    <Status>Confirmed</Status>
    <UserID>Buyer1</UserID>
    <UserIDChanged>false</UserIDChanged>
    <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
    <VATStatus>NoVATTax</VATStatus>
    <BuyerInfo>
      <ShippingAddress>
        <Country>US</Country>
        <PostalCode>95125</PostalCode>
      </ShippingAddress>
    </BuyerInfo>
    <UserAnonymized>false</UserAnonymized>
  </HighBidder>
  <MinimumToBid currencyID="USD">1.25</MinimumToBid>
  <QuantitySold>0</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>false</SecondChanceEligible>
  <ListingStatus>Active</ListingStatus>
</SellingStatus>

<ShippingDetails>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>P4DT23H58M42S</TimeLeft>
<Title>Test Bid{Placed,Received} notify</Title>
<HitCount>0</HitCount>
<GetItFast>false</GetItFast>
<PostalCode>
</PostalCode>
<PictureDetails>
  <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 49

BidReceived

eBay sends a **BidReceived** notification to sellers subscribed whenever a buyer places a bid on a seller's auction item. The typical flow of events is:

- The seller subscribes to the **BidReceived** notification.
- A buyer bids on an auction item.
- eBay sends the **BidReceived** notification to the seller, indicating a buyer just bid on the seller's auction item.

eBay also sends **BidReceived** notifications to the seller for all auction items a buyer bids on, either through the eBay sites or programmatically from an API or SDK.

Data Fields Returned

The data fields the **BidReceived** notification returns are the same as those returned by **GetItemResponse** with the default detail specified.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

In this example, the seller MagicalBookSeller is notified that buyer BountifulBuyer has bid on a copy of “Harry Potter and the Philosopher’s Stone” listed on eBay.

Example 49-1 Sample BidReceived Notification

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
  <ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents" xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
    <ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">pwMi7Lr+Rv3O5CY1geIzSg==</ebl:NotificationSignature>
  </ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
  <GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2011-01-06T08:08:21.443Z</Timestamp>
    <Ack>Success</Ack>
    <CorrelationID>476457110</CorrelationID>
    <Version>699</Version>
    <Build>E699_CORE_BUNDLED_12457306_R1</Build>
    <NotificationEventName>BidReceived</NotificationEventName>
  </GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Example 49-1 Sample BidReceived Notification

```

<RecipientUserID>testuser_magicalbookseller</RecipientUserID>
<EIASToken>nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wFk4CpDZSLpqdjk6x9nY+seQ**</EIASToken>
<Item>
  <AutoPay>false</AutoPay>
  <BuyerProtection>ItemIneligible</BuyerProtection>
  <BuyItNowPrice currencyID="USD">0.0</BuyItNowPrice>
  <Country>US</Country>
  <Currency>USD</Currency>
  <GiftIcon>0</GiftIcon>
  <HitCounter>NoHitCounter</HitCounter>
  <ItemID>110066229559</ItemID>
  <ListingDetails>
    <Adult>false</Adult>
    <BindingAuction>false</BindingAuction>
    <CheckoutEnabled>true</CheckoutEnabled>
    <ConvertedBuyItNowPrice currencyID="USD">0.0</ConvertedBuyItNowPrice>
    <ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
    <ConvertedReservePrice currencyID="USD">0.0</ConvertedReservePrice>
    <HasReservePrice>false</HasReservePrice>
    <StartTime>2011-01-06T08:08:17.000Z</StartTime>
    <EndTime>2011-01-13T08:08:17.000Z</EndTime>
    <ViewItemURL>http://cgi.sandbox.ebay.com/ws/eBayISAPI.dll?ViewItem&item=110066229559&category=377</ViewItemURL>
    <HasUnansweredQuestions>false</HasUnansweredQuestions>
    <HasPublicMessages>false</HasPublicMessages>
    <ExpressListing>false</ExpressListing>
  </ListingDetails>
  <ListingDuration>Days_7</ListingDuration>
  <ListingType>Chinese</ListingType>
  <Location>San Jose, California</Location>
  <PaymentMethods>PayPal</PaymentMethods>
  <PayPalEmailAddress>magicalbookseller@yahoo.com</PayPalEmailAddress>
  <PrimaryCategory>
    <CategoryID>377</CategoryID>
    <CategoryName>Books:Fiction & Literature</CategoryName>
  </PrimaryCategory>
  <PrivateListing>false</PrivateListing>
  <Quantity>1</Quantity>
  <ReservePrice currencyID="USD">0.0</ReservePrice>
  <ReviseStatus>
    <ItemRevised>false</ItemRevised>
  </ReviseStatus>
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>testuser_magicalbookseller@yahoo.com</Email>
  <FeedbackScore>31</FeedbackScore>
  <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Yellow</FeedbackRatingStar>
  <IDVerified>true</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2010-01-01T00:00:00.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>testuser_magicalbookseller</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2007-11-29T18:43:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
<SellerInfo>
  <AllowPaymentEdit>true</AllowPaymentEdit>
  <CheckoutEnabled>true</CheckoutEnabled>
  <CIPBankAccountStored>false</CIPBankAccountStored>
  <GoodStanding>true</GoodStanding>
  <LiveAuctionAuthorized>false</LiveAuctionAuthorized>
  <MerchandizingPref>OptIn</MerchandizingPref>
  <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
  <StoreOwner>true</StoreOwner>
  <StoreURL>http://www.stores.sandbox.ebay.com/id=132854966</StoreURL>
  <ExpressEligible>false</ExpressEligible>
  <ExpressWallet>false</ExpressWallet>
  <SafePaymentExempt>false</SafePaymentExempt>
</SellerInfo>

```

Example 49-1 Sample BidReceived Notification

```
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>testuser_magicalbookseller@yahoo.com</Email>
  <FeedbackScore>31</FeedbackScore>
  <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Yellow</FeedbackRatingStar>
  <IDVerified>true</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2010-01-01T00:00:00.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>testuser_magicalbookseller</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2007-11-29T18:43:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <LiveAuctionAuthorized>false</LiveAuctionAuthorized>
    <MerchandizingPref>OptIn</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <StoreOwner>true</StoreOwner>
    <StoreURL>http://www.stores.sandbox.ebay.com/id=132854966</StoreURL>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
    <SafePaymentExempt>false</SafePaymentExempt>
  </SellerInfo> <MotorsDealer>false</MotorsDealer>
</Seller>

<SellingStatus>
  <BidCount>1</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.0</CurrentPrice>
  <HighBidder>
    <AboutMePage>false</AboutMePage>
    <EIASToken>nY+shZ2PrBndj6wVnY+sEZ2PrA2dj6wFk4CoDZKDPQqdj6x9nY+seQ==</EIASToken>
    <Email>bountifulbuyer@gmail.com</Email>
    <FeedbackScore>32</FeedbackScore>
    <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
    <FeedbackPrivate>false</FeedbackPrivate>
    <FeedbackRatingStar>Yellow</FeedbackRatingStar>
    <IDVerified>false</IDVerified>
    <eBayGoodStanding>true</eBayGoodStanding>
    <NewUser>false</NewUser>
    <RegistrationDate>1995-01-01T00:00:00.000Z</RegistrationDate>
    <Site>US</Site>
    <Status>Confirmed</Status>
    <UserID>testuser_bountifulbuyer</UserID>
    <UserIDChanged>false</UserIDChanged>
    <UserIDLastChanged>2009-02-12T05:01:49.000Z</UserIDLastChanged>
    <VATStatus>NoVATTax</VATStatus>
    <BuyerInfo>
      <ShippingAddress>
        <Country>US</Country>
        <PostalCode>95842</PostalCode>
      </ShippingAddress>
    </BuyerInfo>
    <UserAnonymized>false</UserAnonymized>
  </HighBidder>
  <LeadCount>0</LeadCount>
<MinimumToBid currencyID="USD">1.25</MinimumToBid>
<QuantitySold>0</QuantitySold>
<ReserveMet>true</ReserveMet>
<SecondChanceEligible>false</SecondChanceEligible>
<ListingStatus>Active</ListingStatus>
</SellingStatus>
<ShippingDetails>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
```

Example 49-1 Sample BidReceived Notification

```
<SalesTax>
  <SalesTaxPercent>0.0</SalesTaxPercent>
  <ShippingIncludedInTax>false</ShippingIncludedInTax>
</SalesTax>
<ShippingServiceOptions>
  <ShippingService>USPSMedia</ShippingService>
  <ShippingServiceCost currencyID="USD">2.5</ShippingServiceCost>
  <ShippingServicePriority>1</ShippingServicePriority>
  <ExpeditedService>false</ExpeditedService>
  <ShippingTimeMin>2</ShippingTimeMin>
  <ShippingTimeMax>9</ShippingTimeMax>
</ShippingServiceOptions>
<ShippingType>Flat</ShippingType>
<ThirdPartyCheckout>false</ThirdPartyCheckout>
<ExcludeShipToLocation>Asia</ExcludeShipToLocation>
<ExcludeShipToLocation>Middle East</ExcludeShipToLocation>
<ExcludeShipToLocation>Southeast Asia</ExcludeShipToLocation>
<SellerExcludeShipToLocationsPreference>true</SellerExcludeShipToLocationsPreference>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<Storefront>
  <StoreCategoryID>1</StoreCategoryID>
  <StoreCategory2ID>0</StoreCategory2ID>
  <StoreURL>http://www.stores.sandbox.ebay.com/id=132854966</StoreURL>
</Storefront>
<TimeLeft>P6DT23H59M56S</TimeLeft>
<Title>Harry Potter and the Philosopher's Stone</Title>
<UUID>C59F556B3B634E8EB6638A96928D448D</UUID>
<HitCount>0</HitCount>
<LocationDefaulted>true</LocationDefaulted>
<GetItFast>false</GetItFast>
<PostalCode>95125</PostalCode>
<PictureDetails>
  <GalleryType>Gallery</GalleryType>
  <GalleryURL>http://i1.sandbox.ebayimg.com/03/i/00/61/35/e7_1.JPG?set_id=8800005007</GalleryURL>
  <PhotoDisplay>PicturePack</PhotoDisplay>
  <PictureURL>http://i1.sandbox.ebayimg.com/03/i/00/61/35/e7_1.JPG?set_id=8800005007</PictureURL>
</PictureDetails>
<DispatchTimeMax>3</DispatchTimeMax>
<ProxyItem>false</ProxyItem>
<BuyerGuaranteePrice currencyID="USD">20000.0</BuyerGuaranteePrice>
<IntangibleItem>false</IntangibleItem>
<ReturnPolicy>
  <RefundOption>MoneyBack</RefundOption>
  <Refund>Money Back</Refund>
  <>ReturnsWithinOption>Days_30</ReturnsWithinOption>
  <ReturnsWithin>30 Days</ReturnsWithin>
  <>ReturnsAcceptedOption>ReturnsAccepted</ReturnsAcceptedOption>
  <ReturnsAccepted>Returns Accepted</ReturnsAccepted>
  <Description>This is the first book in the Harry Potter series. In excellent condition!</Description>
  <ShippingCostPaidByOption>Buyer</ShippingCostPaidByOption>
  <ShippingCostPaidBy>Buyer</ShippingCostPaidBy>
</ReturnPolicy>
<PaymentAllowedSite>eBayMotors</PaymentAllowedSite>
<PaymentAllowedSite>CanadaFrench</PaymentAllowedSite>
<PaymentAllowedSite>Canada</PaymentAllowedSite>
<PaymentAllowedSite>US</PaymentAllowedSite>
<ConditionID>3000</ConditionID>
  <ConditionDisplayName>Like New</ConditionDisplayName>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

```
<ShippingServiceOptions>
    <ShippingService>USPSPriority</ShippingService>
    <ShippingServicePriority>2</ShippingServicePriority>
    <ExpeditedService>false</ExpeditedService>
    <ShippingTimeMin>2</ShippingTimeMin>
    <ShippingTimeMax>3</ShippingTimeMax>
</ShippingServiceOptions>
<ShippingType>Calculated</ShippingType>
    <ThirdPartyCheckout>false</ThirdPartyCheckout>
    <ExcludeShipToLocation>Asia</ExcludeShipToLocation>
    <ExcludeShipToLocation>Middle East</ExcludeShipToLocation>
    <ExcludeShipToLocation>Southeast Asia</ExcludeShipToLocation>
    <SellerExcludeShipToLocationsPreference>true</SellerExcludeShipToLocationsPreference>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<Storefront>
    <StoreCategoryID>1</StoreCategoryID>
    <StoreCategory2ID>0</StoreCategory2ID>
    <StoreURL>http://www.stores.sandbox.ebay.com/id=132854966</StoreURL>
</Storefront>
<TimeLeft>P6DT23H59M42S</TimeLeft>
<Title>Harry Potter and the Philosopher's Stone</Title>
<UUID>CD751A9973E249ABB880E1B2DF65B50B</UUID>
<HitCount>0</HitCount>
<LocationDefaulted>true</LocationDefaulted>
<GetItFast>false</GetItFast>
<PostalCode>95125</PostalCode>
<PictureDetails>
    <GalleryType>Gallery</GalleryType>
    <GalleryURL>http://i1.sandbox.ebayimg.com/03/i/00/3e/60/d7_1.JPG?set_id=8800005007</GalleryURL>
    <PhotoDisplay>PicturePack</PhotoDisplay>
    <PictureURL>http://i1.sandbox.ebayimg.com/03/i/00/3e/60/d7_1.JPG?set_id=8800005007</PictureURL>
</PictureDetails>
<DispatchTimeMax>3</DispatchTimeMax>
<ProxyItem>false</ProxyItem>
<BuyerGuaranteePrice currencyID="USD">20000.0</BuyerGuaranteePrice>
<IntangibleItem>false</IntangibleItem>
<ReturnPolicy>
    <RefundOption>MoneyBack</RefundOption>
    <Refund>Money Back</Refund>
    <>ReturnsWithinOption>Days_30</ReturnsWithinOption>
    <ReturnsWithin>30 Days</ReturnsWithin>
    <ReturnsAcceptedOption>ReturnsAccepted</ReturnsAcceptedOption>
    <ReturnsAccepted>Returns Accepted</ReturnsAccepted>
    <Description>This is the first book in the Harry Potter series. In excellent condition!</Description>
    <ShippingCostPaidByOption>Buyer</ShippingCostPaidByOption>
    <ShippingCostPaidBy>Buyer</ShippingCostPaidBy>
</ReturnPolicy>
<PaymentAllowedSite>eBayMotors</PaymentAllowedSite>
<PaymentAllowedSite>CanadaFrench</PaymentAllowedSite>
<PaymentAllowedSite>Canada</PaymentAllowedSite>
<PaymentAllowedSite>US</PaymentAllowedSite>
    <ConditionID>1000</ConditionID>
    <ConditionDisplayName>Brand New</ConditionDisplayName>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 50

BulkDataExchangeJobCompleted

eBay sends a **BulkDataExchangeJobCompleted** notification when the eBay Bulk Data Exchange server has finished processing a Bulk Data Exchange job. A registered Large Merchant Services user must subscribe to this notification using the [SetNotificationPreferences](#) call in the Trading API.

Platform Choice

When a seller requests, via the [SetNotificationPreferences](#) call, to be notified of this event, the default is for the notification to be sent via the Trading API ("Platform"). The user can choose instead to be notified via wireless ("SMS").

Data Fields Returned

The returned jobProfile fields are the same as those returned by the getJobStatus call. They are shown in the following table.

Table 50-1

Field	Definition
<code>jobProfile.jobId</code>	The JobID obtained by calling <code>createUploadJob</code> .
<code>jobProfile.jobType</code>	Specifies the Merchant Data API call used in a Bulk Data Exchange service job.
<code>jobProfile.jobStatus</code>	Returns the current state of a Bulk Data Exchange Job. If you are receiving this notification, the value in this field should be, "Completed".
<code>jobProfile.completionTime</code>	The date and time that job processing was completed.
<code>jobProfile.errorCount</code>	The number of non-fatal (informational) errors that occurred during file processing (uploads or reports) after you sent the Bulk Data Exchange <code>startUploadJobRequest</code> call or the <code>startDownloadJobRequest</code> call.
<code>jobProfile.percentComplete</code>	The percentage of your file that has been processed after sending the Bulk Data Exchange <code>startUploadJobRequest</code> call or the <code>startDownloadJobRequest</code> call. If you are receiving this notification, the value in this field should be "100.0", as shown in the example that follows.
<code>jobProfile.fileReferenceId</code>	Time the question was created (the time it was sent).

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 50-1 Sample BulkDataExchangeJobCompleted Notification

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header/>
  <soapenv:Body>
    <getJobStatusResponse xmlns:sct="http://www.ebay.com/soaframework/common/types" xmlns="http://www.ebay.com/marketplace/services">
      <ack>Success</ack>
      <version>1.0.0</version>
      <timestamp>2008-11-05T03:37:19.304Z</timestamp>
      <jobProfile>
        <jobId>50000000266</jobId>
        <jobType>AddFixedPriceItem</jobType>
        <jobStatus>Completed</jobStatus>
        <creationTime>2008-11-04T21:58:04.000Z</creationTime>
        <completionTime>2008-11-04T22:01:46.000Z</completionTime>
        <errorCount>0</errorCount>
        <percentComplete>100.0</percentComplete>
        <fileReferenceId>50000000506</fileReferenceId>
      </jobProfile>
    </getJobStatusResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Chapter 51

Checkout

The **Checkout** notification goes out for items purchased on eBay. Characteristics of this notification:

- Contains the same data fields as the [AuctionCheckoutComplete](#) notification and the [GetItemTransactions](#) call at the default detail level. These include:
 - Buyer information (shipping address, email address)
 - Payment/Shipping information (shipping service address, amounts paid)
 - etc.
- Transmits transaction data on a single, particular transaction in the eBay system
- Is sent only for transactions for items that are purchased on eBay and have [ThirdPartyCheckoutIntegration](#) set to true
 - See the [TransactionPlatform](#) field to determine whether an item was purchased on eBay.
- Is sent several times per transaction
- Only includes an order ID if the transaction in question is part of an *order* (a combined payment for multiple purchases from the same seller). This enables notification recipients to identify the order applicable to the transaction and potentially process on a per-order basis. For information about orders, see "[Creating a Combined Payment Order](#)" on page 412.

No [OrderID](#) will appear in the Checkout notification when only 1 item is purchased from the same seller in a single end-to-end checkout flow (because the transaction will not be part of an order in that case).

- See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Checkout is sent the first time for every transaction created for a [ThirdPartyCheckoutIntegration](#) item. If the transaction is part of an order, the **Checkout** notification will include an order ID. However, each transaction in an order will generate a separate **Checkout** notification. Note that at this point, the payment has been authorized for this transaction/order but has not been settled (that is, funds have not been moved).

Checkout is sent a second time when the settlement of the transaction is completed. If settlement fails, **Checkout** is sent as soon as any immediate processing of the transaction is completed (e.g. the transaction is canceled).

Checkout may also be sent after a [ReviseCheckoutStatus](#) call updates the status of a [ThirdPartyCheckoutIntegration](#) transaction. This may occur after the settlement of the transaction

(second notification above) but will only occur in rare cases, such as when a seller elects to reject a shipment to a PayPal unconfirmed address.

Example

Example 51-1 Sample Checkout Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ebl:RequesterCredentials soapenv:mustUnderstand="0"
      xmlns:ns="urn:ebay:apis:eBLBaseComponents" xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
      <ebl:NotificationSignature
        xmlns:ebl="urn:ebay:apis:eBLBaseComponents">AKurONibBT6zkNCsQ27rtg==</ebl:NotificationSignature>
    </ebl:RequesterCredentials>
  </soapenv:Header>
  <soapenv:Body>
    <GetItemTransactionsResponse xmlns="urn:ebay:apis:eBLBaseComponents">
      <Timestamp>2006-4-27T17:48:03.545Z</Timestamp>
      <Ack>Success</Ack>
      <CorrelationID>12345678</CorrelationID>
      <Version>455</Version>
      <Build>20060426141851</Build>
      <NotificationEventName>Checkout</NotificationEventName>
      <PaginationResult>
        <TotalNumberOfPages>1</TotalNumberOfPages>
        <TotalNumberOfEntries>1</TotalNumberOfEntries>
      </PaginationResult>
      <HasMoreTransactions>false</HasMoreTransactions>
      <TransactionsPerPage>100</TransactionsPerPage>
      <PageNumber>1</PageNumber>
      <ReturnedTransactionCountActual>1</ReturnedTransactionCountActual>
      <Item>
        <AutoPay>false</AutoPay>
        <BuyerProtection>ItemIneligible</BuyerProtection>
        <Currency>USD</Currency>
        <ItemID>2211777403</ItemID>
        <ListingType>Chinese</ListingType>
        <PaymentMethods>PaymentSeeDescription</PaymentMethods>
        <PrivateListing>true</PrivateListing>
        <Quantity>1</Quantity>
        <Seller>
          <AboutMePage>true</AboutMePage>
          <EIASToken>nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wJnY+gD5iFoA+dj6x9nY+seQ==</EIASToken>
          <Email>sampleseller@sampleseller.com</Email>
          <FeedbackScore>4119</FeedbackScore>
          <FeedbackRatingStar>Red</FeedbackRatingStar>
          <IDVerified>true</IDVerified>
          <NewUser>false</NewUser>
          <RegistrationDate>2004-10-03T18:28:52.000Z</RegistrationDate>
          <Site>US</Site>
          <Status>Confirmed</Status>
          <UserID>sampleseller</UserID>
          <UserIDChanged>false</UserIDChanged>
          <UserIDLastChanged>2001-01-19T17:21:01.000Z</UserIDLastChanged>
        </Seller>
      </Item>
    </GetItemTransactionsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 51-1 Sample Checkout Notification

```
<SellerInfo>
  <AllowPaymentEdit>true</AllowPaymentEdit>
  <CheckoutEnabled>true</CheckoutEnabled>
  <CIPBankAccountStored>false</CIPBankAccountStored>
  <GoodStanding>true</GoodStanding>
  <MerchandizingPref>OptIn</MerchandizingPref>
  <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
  <SellerLevel>None</SellerLevel>
  <StoreOwner>true</StoreOwner>
  <StoreURL>http://www.ebaystores.ebay.com/id=178713</StoreURL>
</SellerInfo>
</Seller>
<SellingStatus>
  <ConvertedCurrentPrice currencyID="USD">100.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">100.0</CurrentPrice>
  <QuantitySold>1</QuantitySold>
</SellingStatus>
<Site>US</Site>
<Title>GetItem Test</Title>
</Item>
<TransactionArray>
  <Transaction>
    <AmountPaid currencyID="USD">100.0</AmountPaid>
    <AdjustmentAmount currencyID="USD">0.0</AdjustmentAmount>
    <ConvertedAdjustmentAmount currencyID="USD">0.0</ConvertedAdjustmentAmount>
    <Buyer>
      <AboutMePage>false</AboutMePage>
      <EIASToken>nY+sHZ2PrBndj6wVnY+sEZ2PrA2dj6wJnY+gD5iFogidj6x9nY+seQ==</EIASToken>
      <Email>samplebuyer@samplebuyer.com</Email>
      <FeedbackScore>58</FeedbackScore>
      <FeedbackRatingStar>Blue</FeedbackRatingStar>
      <IDVerified>true</IDVerified>
      <NewUser>false</NewUser>
      <RegistrationDate>1999-01-19T08:00:00.000Z</RegistrationDate>
      <Site>US</Site>
      <Status>Confirmed</Status>
      <UserID>samplebuyer</UserID>
      <UserIDChanged>false</UserIDChanged>
      <UserIDLastChanged>2001-01-19T18:42:14.000Z</UserIDLastChanged>
    <BuyerInfo>
      <ShippingAddress>
        <Name>new_Name</Name>
        <Street1>a new street1</Street1>
        <Street2>new street 2</Street2>
        <CityName>my new town</CityName>
        <StateOrProvince>CA</StateOrProvince>
        <Country>US</Country>
        <CountryName>United States</CountryName>
        <Phone>(111) 111-1</Phone>
        <PostalCode>95555</PostalCode>
        <AddressID>191925</AddressID>
        <AddressOwner>eBay</AddressOwner>
      </ShippingAddress>
    </BuyerInfo>
  </Buyer>
</Transaction>
</TransactionArray>
```

Example 51-1 Sample Checkout Notification

```
<ShippingDetails>
  <ChangePaymentInstructions>true</ChangePaymentInstructions>
  <InsuranceFee currencyID="USD">0.0</InsuranceFee>
  <InsuranceOption>NotOffered</InsuranceOption>
  <InsuranceWanted>false</InsuranceWanted>
  <PaymentEdited>false</PaymentEdited>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ShippingServiceOptions>
    <ShippingService>ShippingMethodStandard</ShippingService>
    <ShippingServiceCost currencyID="USD">0.0</ShippingServiceCost>
    <ShippingServiceAdditionalCost currencyID="USD">0.0
      </ShippingServiceAdditionalCost>
    <ShippingServicePriority>1</ShippingServicePriority>
  </ShippingServiceOptions>
</ShippingDetails>
<ConvertedAmountPaid currencyID="USD">100.0</ConvertedAmountPaid>
<ConvertedTransactionPrice currencyID="USD">100.0</ConvertedTransactionPrice>
<CreatedDate>2006-4-09T01:27:52.000Z</CreatedDate>
<DepositType>None</DepositType>
<QuantityPurchased>1</QuantityPurchased>
<Status>
  <eBayPaymentStatus>NoPaymentFailure</eBayPaymentStatus>
  <CheckoutStatus>CheckoutComplete</CheckoutStatus>
  <LastTimeModified>2006-4-27T17:46:04.000Z</LastTimeModified>
  <PaymentMethodUsed>PaymentSeeDescription</PaymentMethodUsed>
  <CompleteStatus>Complete</CompleteStatus>
</Status>
<TransactionID>0</TransactionID>
<TransactionPrice currencyID="USD">100.0</TransactionPrice>
<BestOfferSale>false</BestOfferSale>
<ShippingServiceSelected>
  <ShippingInsuranceCost currencyID="USD">0.0</ShippingInsuranceCost>
  <ShippingService>ShippingMethodStandard</ShippingService>
  <ShippingServiceCost currencyID="USD">0.0</ShippingServiceCost>
</ShippingServiceSelected>
</Transaction>
</TransactionArray>
<PayPalPreferred>true</PayPalPreferred>
</GetItemTransactionsResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 52

CounterOfferReceived

eBay sends a **CounterOfferReceived** notification to third parties subscribed on the buyer's behalf whenever a buyer receives a counter offer from the seller in response to a Best Offer. The typical flow of events is:

- A third party subscribes to the **CounterOfferReceived** notification on behalf of a buyer.
- The buyer receives a counter-offer from a seller in response to a Best Offer the buyer placed.
- eBay sends the **CounterOfferReceived** notification to the third party, indicating the buyer received a counter-offer.

eBay also sends **CounterOfferReceived** notifications to the third party for all counter offers the buyer receives, either through the eBay sites or programmatically from an API or SDK.

Data Fields Returned

The data fields the **CounterOfferReceived** notification returns are the same as those returned by **GetBestOfferResponse** with the **ReturnAll** detail level specified.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 52-1 Sample CounterOfferReceived Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
      xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
      <ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">fbKLYvy5DGxXn1P8v0mSuw==</
      ebl:NotificationSignature>
    </ebl:RequesterCredentials>
  </soapenv:Header>
  <soapenv:Body>
    <GetBestOffersResponse xmlns="urn:ebay:apis:eBLBaseComponents">
      <Timestamp>2008-02-13T03:21:32.606Z</Timestamp>
      <Ack>Success</Ack>
      <CorrelationID>137540710</CorrelationID>
      <Version>553</Version>
      <Build>20050429131630</Build>
      <NotificationEventName>CounterOfferReceived</NotificationEventName>
      <RecipientUserID>Buyer1</RecipientUserID>
      <BestOfferArray>
        <BestOffer>
          <BestOfferID>6718</BestOfferID>
          <ExpirationTime>2008-02-15T03:21:19.000Z</ExpirationTime>
          <Buyer>
            <Email>buyer@email.com</Email>
            <FeedbackScore>1</FeedbackScore>
            <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
            <UserID>Buyer1</UserID>
          </Buyer>
          <Price currencyID="USD">200.0</Price>
          <Status>Pending</Status>
          <Quantity>1</Quantity>
          <BestOfferCodeType>SellerCounterOffer</BestOfferCodeType>
        </BestOffer>
      </BestOfferArray>
      <Item>
        <BuyItNowPrice currencyID="USD">1000.0</BuyItNowPrice>
        <ItemID>250000627087</ItemID>
        <ListingDetails>
          <EndTime>2008-02-18T03:20:32.000Z</EndTime>
        </ListingDetails>
        <Location>San Jose, CA</Location>
        <Title>Test CounterOfferReceived</Title>
      </Item>
    </GetBestOffersResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Chapter 53

CheckoutBuyerRequestsTotal

The **CheckoutBuyerRequestsTotal** notification is sent to the seller every time the buyer goes through the process of requesting a total from the seller. This event occurs before checkout is complete.

For fixed-price items, **CheckoutBuyerRequestsTotal** occurs before **FixedPriceTransaction**. For auction items, **CheckoutBuyerRequestsTotal** occurs after **EndOfAuction**, but before **AuctionCheckoutComplete**.

A **CheckoutBuyerRequestsTotal** notification may be sent multiple times for the same auction if a buyer requests a total more than once.

Data Fields Returned

The **CheckoutBuyerRequestsTotal** notification returns the same data as the **GetItemTransactions** call with the default detail level.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Chapter 54

eBay Buyer Protection Notifications

- “eBP Notification Structure” on page 667
- “EBPMyResponseDue” on page 667
- “EBPOtherPartyResponseDue” on page 668
- “EBPEscalatedCase” on page 668
- “EBPAppealedCase” on page 668
- “EBPMyPaymentDue” on page 668
- “EBPMyPaymentDone” on page 668
- “EBPClosedAppeal” on page 669
- “EBPClosedCase” on page 669
- “EBPOnHoldCase” on page 669

With eBay Buyer Protection notifications, buyers and sellers that are part of an eBay Buyer Protection (eBP) case can be notified when the case changes state. The two case types that are created through eBay’s Resolution Center are Item Not Received and Significantly Not As Described cases. The following event notifications can be subscribed to and sent to the buyer or seller for eBay Buyer Protection cases:

- **EBPMyResponseDue:** notification sent to the buyer or seller when a response to the case is due from the call user. When a case is opened, this notification is only sent to the seller involved in the case and not the buyer.
- **EBPOtherPartyResponseDue:** notification sent to the buyer or seller when a response to the case is due from the other party in the case.
- **EBPEscalatedCase:** notification sent to the buyer or seller when a case is escalated.
- **EBPAppealedCase:** notification sent to the buyer or seller when the decision of a case is appealed.
- **EBPMyPaymentDue:** notification sent to the seller when payment (to eBay or buyer) is due.
- **EBPMyPaymentDone:** notification sent to the seller when payment (to eBay or buyer) is processed.
- **EBPClosedAppeal:** notification sent to the buyer or seller when an appeal of a case decision has been closed.
- **EBPClosedCase:** notification sent to the buyer or seller when a case has been closed.
- **EBPOnHoldCase:** notification sent to the buyer or seller when a case has been placed on hold.

These notifications are set in the [SetNotificationPreferences](#) call. Before using these notifications, be sure that you have read and understood the information in “[Working with Platform Notifications](#)” on page 623.

eBP Notification Structure

The following sample shows the payload for an [EBPMyResponseDue](#) notification.

Example 54-1 Sample EBPMyResponseDue Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://
www.ebay.com/marketplace/resolution/v1/services">
<soapenv:Header/>
<soapenv:Body>
<NotificationEvent>
<NotificationEventName>EBPMyResponseDue</NotificationEventName>
<RecipientUserID>cps_seller3</RecipientUserID>
<ExternalUserData>AABBCC .. </ExternalUserData>
<CaseId>5000004995</CaseId>
<CaseType>EBP_INR</CaseType>
<ItemId>280001402455</ItemId>
<TransactionId>7468917018</TransactionId>
<CreationDate>2009-09-07T12:50:06.333Z</CreationDate>
<OtherPartyId>cps_buyer3</OtherPartyId>
<OtherPartyRole>BUYER</OtherPartyRole>
<CaseStatus>MY_RESPONSE_DUE</CaseStatus>
<CaseGlobalId>EBAY-US</CaseGlobalId>
</NotificationEvent>
</soapenv:Body>
</soapenv:Envelope>
```

The payload structure for each eBay Buyer Protection notification is essentially the same. The key fields are described below:

- **NotificationEventName:** the name of the notification
- **RecipientUserID:** the eBay user ID for the notification subscriber.
- **ExternalUserData:** this field is pulled from the [SetNotificationPreferences](#) request. See the [SetNotificationPreferences](#) call reference for more information about this field.

For more information on other fields in the payload, see the [getEBPCaseDetail](#) call reference documentation in the Resolution Case Management API documentation set.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

EBPMyResponseDue

The [EBPMyResponseDue](#) notification is sent to the subscriber when their response is due on a case. This notification can be sent to the buyer or seller, except it is only sent to the seller when a new case is opened by a buyer against that seller. The enumeration value for this case activity (returned in the [getEBPCaseDetail](#) call of the Resolution Case Management API) is MY_RESPONSE_DUE. If viewed in the eBay Resolution Center, the case record will indicate whether the response is due from the buyer or seller.

EBPOtherPartyResponseDue

The **EBPOtherPartyResponseDue** notification is sent to the subscriber when the response of the other party is due on a case. This notification can be sent to the buyer or seller. The enumeration value for this case activity (returned in the [getEBPCaseDetail](#) call of the Resolution Case Management API) is OTHER_PARTY_RESPONSE_DUE. If viewed in the eBay Resolution Center, the case record will indicate whether the response is due from the buyer or seller.

EBPEscalatedCase

The **EBPEscalatedCase** notification is sent to the subscriber when a case has been escalated. This notification can be sent to the buyer or seller. The enumeration value for this case activity (returned in the [getEBPCaseDetail](#) call of the Resolution Case Management API) will be either YOU_CONTACTED_CS_AWAITING_RESPONSE or OTHER_PARTY_CONTACTED_CS_AWAITING_RESPONSE, based on whom escalated the case. If viewed in the eBay Resolution Center, the case record will indicate whether the case was escalated by the buyer or seller.

EBPAppealedCase

The **EBPAppealedCase** notification is sent to the subscriber when the decision of a case has been appealed. This notification can be sent to the buyer or seller. The enumeration value for this case activity (returned in the [getEBPCaseDetail](#) call of the Resolution Case Management API) is YOU_CONTACTED_CS_ABOUT_CLOSED_CASE. If viewed in the eBay Resolution Center, the case record will indicate whether an appeal was made by the buyer or seller.

EBPMypaymentdue

The **EBPMypaymentdue** notification is sent to the subscribed seller when a payment is due on a case. This notification can only be sent to the seller. The enumeration value for this case activity (returned in the [getEBPCaseDetail](#) call of the Resolution Case Management API) is MY_PAYMENT_DUE. If viewed in the eBay Resolution Center, the case record will indicate whether the payment due is for a standard case or for a case on which there was an appeal.

EBPMypaymentdone

The **EBPMypaymentdone** notification is sent to the subscribed seller when a payment has been processed on a case. This notification can only be sent to the seller. The enumeration value for this case activity (returned in the [getEBPCaseDetail](#) call of the Resolution Case Management API) is PAID. If viewed in the eBay Resolution Center, the case record will indicate whether the processed payment was for a standard case or for a case on which there was an appeal.

EBPClosedAppeal

The [EBPClosedAppeal](#) notification is sent to the subscriber when the appeal related to a case has been closed. This notification can be sent to the buyer or seller. The enumeration value for this case activity (returned in the [getEBPCaseDetail](#) call of the Resolution Case Management API) is CS_CLOSED. If viewed in the eBay Resolution Center, the case record will indicate whether the appeal was made by the buyer or seller.

EBPClosedCase

The [EBPClosedCase](#) notification is sent to the subscriber when a case has been closed. This notification can be sent to the buyer or seller. The enumeration value for this case activity (returned in the [getEBPCaseDetail](#) call of the Resolution Case Management API) is either CS_CLOSED (if case was escalated) or CLOSED.

EBPOnHoldCase

The [EBPOnHoldCase](#) notification is sent to the subscriber when a case has placed on hold. This notification can be sent to the buyer or seller. The enumeration value for this case activity (returned in the [getEBPCaseDetail](#) call of the Resolution Case Management API) is either CS_CLOSED (if case was escalated) or CLOSED.

Chapter 55

EndOfAuction

An **EndOfAuction** notification is sent when an auction item ends. An auction ends either when its duration expires (whether or not there is a winning bidder), the item is purchased with Buy It Now, or the auction has been cancelled.

Platform Choice

When a user requests, via the notification preferences web page, to be notified of this event, the default is for the notification to be sent via API ("Platform"). The user can choose instead to be notified via wireless ("SMS").

Data Fields Returned

The **EndOfAuction** notification returns the same data as the **GetItemTransactions** call with the **ReturnAll** detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

In this example, MagicalBookSeller is notified that her listing of “Harry Potter and the Philosopher’s Stone” has ended with a successful bid from BountifulBuyer.

Example 55-1 Sample EndOfAuction Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
  www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
  <ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
    xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
    <ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">1w5Fdyr9V9ofTq67etR01A==</
      ebl:NotificationSignature>
    </ebl:RequesterCredentials>
  </soapenv:Header>
<soapenv:Body>
  <GetItemTransactionsResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2011-01-06T08:08:27.715Z</Timestamp>
    <Ack>Success</Ack>
    <CorrelationID>476457120</CorrelationID>
    <Version>699</Version>
    <Build>E699_CORE_BUNDLED_12457306_R1</Build>
    <NotificationEventName>EndOfAuction</NotificationEventName>
    <RecipientUserID>testuser_magicalbookseller</RecipientUserID>
    <EIAToken>nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wFk4CpDZSLpwqdj6x9nY+seQ**</EIAToken>
    <PaginationResult>
      <TotalNumberOfPages>1</TotalNumberOfPages>
      <TotalNumberOfEntries>1</TotalNumberOfEntries>
    </PaginationResult>
    <HasMoreTransactions>false</HasMoreTransactions>
    <TransactionsPerPage>100</TransactionsPerPage>
    <PageNumber>1</PageNumber>
    <ReturnedTransactionCountActual>1</ReturnedTransactionCountActual>
  <Item>
    <AutoPay>false</AutoPay>
    <BuyItNowPrice currencyID="USD">0.0</BuyItNowPrice>
    <Currency>USD</Currency>
    <ItemID>110066229559</ItemID>
    <ListingDetails>
      <RelistedItemID>110066229564</RelistedItemID>
      <StartTime>2011-01-06T08:08:17.000Z</StartTime>
      <EndTime>2011-01-06T08:08:24.000Z</EndTime>
      <ViewItemURL>http://cgi.sandbox.ebay.com/ws/eBayISAPI.dll?ViewItem&Item=110066229559</ViewItemURL>
      <ExpressListing>false</ExpressListing>
      <ViewItemURLForNaturalSearch>http://cgi.sandbox.ebay.com/ws/
        eBayISAPI.dll?ViewItem&item=110066229559&category=0</ViewItemURLForNaturalSearch>
    </ListingDetails>
    <ListingType>Chinese</ListingType>
    <PaymentMethods>PayPal</PaymentMethods>
    <PrimaryCategory>
      <CategoryID>377</CategoryID>
    </PrimaryCategory>
    <PrivateListing>false</PrivateListing>
    <Quantity>1</Quantity>
    <SecondaryCategory>
      <CategoryID>0</CategoryID>
    </SecondaryCategory>
  </Item>
</GetItemTransactionsResponse>

```

Example 55-1 Sample EndOfAuction Notification

```
<Seller>
  <AboutMePage>false</AboutMePage>
  <EIASToken>nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wFk4CpDZSLpwqjd6x9nY+seQ==</EIASToken>
  <Email>testuser_magicalbookseller@yahoo.com</Email>
  <FeedbackScore>31</FeedbackScore>
  <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Yellow</FeedbackRatingStar>
  <IDVerified>true</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2010-01-01T00:00:00.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>testuser_magicalbookseller</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2007-11-29T18:43:54.000Z</UserIDLastChanged>

<VATStatus>NoVATTax</VATStatus>
<SellerInfo>
  <AllowPaymentEdit>true</AllowPaymentEdit>
  <CheckoutEnabled>true</CheckoutEnabled>
  <CIPBankAccountStored>false</CIPBankAccountStored>
  <GoodStanding>true</GoodStanding>
  <LiveAuctionAuthorized>false</LiveAuctionAuthorized>
  <MerchandizingPref>Opt In</MerchandizingPref>
  <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
  <StoreOwner>true</StoreOwner>
  <StoreURL>http://www.stores.sandbox.ebay.com/id=132854966</StoreURL>
  <ExpressEligible>false</ExpressEligible>
  <ExpressWallet>false</ExpressWallet>
  <SafePaymentExempt>false</SafePaymentExempt>
</SellerInfo>
</Seller>

<SellingStatus>
  <BidCount>1</BidCount>
  <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.0</CurrentPrice>
  <QuantitySold>1</QuantitySold>
  <ListingStatus>Completed</ListingStatus>
</SellingStatus>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<Title>Harry Potter and the Philosopher's Stone</Title>
<GetItFast>false</GetItFast>
<IntegratedMerchantCreditCardEnabled>false</IntegratedMerchantCreditCardEnabled>
<ConditionID>3000</ConditionID>
<ConditionDisplayName>Like New</ConditionDisplayName>
</Item>
```

Example 55-1 Sample EndOfAuction Notification

```
<TransactionArray>
  <Transaction>
    <AmountPaid currencyID="USD">3.5</AmountPaid>
    <AdjustmentAmount currencyID="USD">0.0</AdjustmentAmount>
    <ConvertedAdjustmentAmount currencyID="USD">0.0</ConvertedAdjustmentAmount>
    <Buyer>
      <AboutMePage>false</AboutMePage>
      <EIAToken>nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wFk4CoDZKDpQqdj6x9nY+seQ==</EIAToken>
      <Email>bountifulbuyer@gmail.com</Email>
      <FeedbackScore>32</FeedbackScore>
      <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
      <FeedbackPrivate>false</FeedbackPrivate>
      <FeedbackRatingStar>Yellow</FeedbackRatingStar>
      <IDVerified>false</IDVerified>
      <eBayGoodStanding>true</eBayGoodStanding>
      <NewUser>false</NewUser>
      <RegistrationDate>1995-01-01T00:00:00.000Z</RegistrationDate>
      <Site>US</Site>
      <Status>Confirmed</Status>
      <UserID>testuser_bountifulbuyer</UserID>
      <UserIDChanged>false</UserIDChanged>
      <UserIDLastChanged>2009-02-12T05:01:49.000Z</UserIDLastChanged>
      <VATStatus>NoVATTax</VATStatus>
      <BuyerInfo>
        <ShippingAddress>
          <Name>Bountiful Buyer</Name>
          <Street1>777 Klamath Drive</Street1>
          <CityName>Sacramento</CityName>
          <StateOrProvince>CA</StateOrProvince>
          <Country>US</Country>
          <CountryName>United States</CountryName>
          <PostalCode>95842</PostalCode>
          <AddressID>5383702</AddressID>
          <AddressOwner>eBay</AddressOwner>
          <AddressUsage>DefaultShipping</AddressUsage>
        </ShippingAddress>
      </BuyerInfo>
      <UserAnonymized>false</UserAnonymized>
    </Buyer>
```

Example 55-1 Sample EndOfAuction Notification

```
<ShippingDetails>
  <ChangePaymentInstructions>true</ChangePaymentInstructions>
  <PaymentEdited>false</PaymentEdited>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
    <SalesTaxAmount currencyID="USD">0.0</SalesTaxAmount>
  </SalesTax>
  <ShippingServiceOptions>
    <ShippingService>USPSMedia</ShippingService>
    <ShippingServiceCost currencyID="USD">2.5</ShippingServiceCost>
    <ShippingServicePriority>1</ShippingServicePriority>
    <ExpeditedService>false</ExpeditedService>
    <ShippingTimeMin>2</ShippingTimeMin>
    <ShippingTimeMax>9</ShippingTimeMax>
  </ShippingServiceOptions>
  <ShippingType>Flat</ShippingType>
  <SellingManagerSalesRecordNumber>1262</SellingManagerSalesRecordNumber>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
  <GetItFast>false</GetItFast>
</ShippingDetails>
<ConvertedAmountPaid currencyID="USD">3.5</ConvertedAmountPaid>
<ConvertedTransactionPrice currencyID="USD">1.0</ConvertedTransactionPrice>
<CreatedDate>2011-01-06T08:08:24.000Z</CreatedDate>
<DepositType>None</DepositType>
<QuantityPurchased>1</QuantityPurchased>
<Status>
  <eBayPaymentStatus>NoPaymentFailure</eBayPaymentStatus>
  <CheckoutStatus>CheckoutIncomplete</CheckoutStatus>
  <LastTimeModified>2011-01-06T08:08:25.000Z</LastTimeModified>
  <PaymentMethodUsed>None</PaymentMethodUsed>
  <CompleteStatus>Incomplete</CompleteStatus>
  <BuyerSelectedShipping>false</BuyerSelectedShipping>
  <PaymentHoldStatus>None</PaymentHoldStatus>
  <IntegratedMerchantCreditCardEnabled>false</IntegratedMerchantCreditCardEnabled>
</Status>
<TransactionID>0</TransactionID>
  <TransactionPrice currencyID="USD">1.0</TransactionPrice>
  <BestOfferSale>false</BestOfferSale>
  <ShippingServiceSelected>
    <ShippingService>USPSMedia</ShippingService>
    <ShippingServiceCost currencyID="USD">2.5</ShippingServiceCost>
  </ShippingServiceSelected>
  <FinalValueFee currencyID="USD">0.09</FinalValueFee>
  <TransactionSiteID>US</TransactionSiteID>
  <Platform>eBay</Platform>
  <PayPalEmailAddress>magicalbookseller@yahoo.com</PayPalEmailAddress>
  <BuyerGuaranteePrice currencyID="USD">20000.0</BuyerGuaranteePrice>
  <IntangibleItem>false</IntangibleItem>
  </Transaction>
</TransactionArray>
<PayPalPreferred>true</PayPalPreferred>
</GetItemTransactionsResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 56

Feedback Notifications

["Feedback" on page 675](#)

Feedback

The **Feedback** notification is sent to the seller when initial feedback comments are left and, for items in dispute, the notification is sent when updated, follow-up comments are left.

The **Feedback** notification is a user-level notification. All applications that subscribe will get the notification, regardless of how the relevant item was listed (i.e., whether the item was listed through the API or through the web).

Data Fields Returned

The **Feedback** notification returns the same data as the [GetFeedback](#) call with the **ReturnAll** detail level. However, although sellers may be able to use [GetFeedback](#) to retrieve the averages of detailed seller ratings (see ["Detailed Seller Ratings" on page 433](#)), this data is not in the **Feedback** notification.

To determine whether the feedback was left for the buyer or the seller, see the [FeedbackDetailRole](#).

See ["SOAP Message Body" on page 628](#) for details on the standard elements returned for all notifications.

Example

Example 56-1 Sample Feedback Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ebl:RequesterCredentials soapenv:mustUnderstand="0">
      xmlns:ns="urn:ebay:apis:eBLBaseComponents" xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
        <ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
          zQrBxqnc9Y1tJaJVoDME7g==</ebl:NotificationSignature>
        </ebl:RequesterCredentials>
    </soapenv:Header>
  <soapenv:Body>
    <GetFeedbackResponse xmlns="urn:ebay:apis:eBLBaseComponents">
      <Timestamp>2005-04-30T03:40:37.529Z</Timestamp>
      <Ack>Success</Ack>
      <CorrelationID>82083320</CorrelationID>
      <Version>407</Version>
      <Build>20050429131630</Build>
      <NotificationEventName>Feedback</NotificationEventName>
      <RecipientUserID>seller1</RecipientUserID>
      <FeedbackDetailArray>
        <FeedbackDetail>
          <CommentingUser>buyer1</CommentingUser>
          <CommentingUserScore>2</CommentingUserScore>
          <CommentText>Your product sucks!</CommentText>
          <CommentTime>2005-04-30T03:39:46.000Z</CommentTime>
          <CommentType>Negative</CommentType>
          <ItemID>704637038</ItemID>
          <Role>Seller</Role>
          <FeedbackID>3900595379</FeedbackID>
          <TransactionID>854387</TransactionID>
        </FeedbackDetail>
      </FeedbackDetailArray>
      <FeedbackDetailItemTotal>151</FeedbackDetailItemTotal>
      <FeedbackScore>-1</FeedbackScore>
    </GetFeedbackResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Chapter 57

FeedbackLeft

eBay sends a **FeedbackLeft** notification to third parties subscribed on the user's behalf whenever initial feedback comments are left by that user and when updated follow-up comments are left for items in dispute. The typical flow of events is:

- 1 A third party subscribes to the **FeedbackLeft** notification on behalf of a user.
- 2 The user leaves initial feedback comments or updated follow-up comments for items in dispute.
- 3 eBay sends the **FeedbackLeft** notification to the third party, indicating that the user left new feedback.

eBay also sends **FeedbackLeft** notifications to the third party for all feedback items the user leaves, either through the eBay sites or programmatically from an API or SDK.

Data Fields Returned

The data fields the **FeedbackLeft** notification returns are the same as those returned by **GetFeedbackResponse** with the **ReturnAll** detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 57-1 Sample FeedbackLeft Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">0xhgztR7iOctAlekC39KcA==</
ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetFeedbackResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-02-13T02:27:42.996Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>137540280</CorrelationID>
<Version>553</Version>
<Build>20050429131630</Build>
<NotificationEventName>FeedbackLeft</NotificationEventName>
<RecipientUserID>Seller1</RecipientUserID>
<FeedbackDetailArray>
<FeedbackDetail>
<CommentingUser>Buyer1</CommentingUser>
<CommentingUserScore>100</CommentingUserScore>
<CommentText>Thanks for the payment!</CommentText>
<CommentTime>2008-02-13T02:27:16.000Z</CommentTime>
<CommentType>Positive</CommentType>
<ItemID>250000627045</ItemID>
<Role>Buyer</Role>
<FeedbackID>11830457019</FeedbackID>
</FeedbackDetail>
</FeedbackDetailArray>
<FeedbackDetailItemTotal>214</FeedbackDetailItemTotal>
<FeedbackScore>1</FeedbackScore>
</GetFeedbackResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 58

FeedbackReceived

eBay sends a **FeedbackReceived** notification to third parties subscribed on the user's behalf whenever initial feedback comments are received by that user and when updated follow-up comments are received for items in dispute. The typical flow of events is:

- 1 A third party subscribes to the **FeedbackReceived** notification on behalf of a user.
- 2 The user receives initial feedback comments or updated follow-up comments for items in dispute.
- 3 eBay sends the **FeedbackReceived** notification to the third party, indicating the user received new feedback.
- 4 eBay also sends **FeedbackReceived** notifications to the third party for all feedback items the user receives, either through the eBay sites or programmatically from an API or SDK.

Data Fields Returned

The data fields the **FeedbackReceived** notification returns are the same as those returned by **GetFeedbackResponse** with the **ReturnAll** detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 58-1 Sample FeedbackReceived Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">iliJPJUzk7mZQMK1Ytrgg/g==</
ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetFeedbackResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-02-13T02:27:45.215Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>137540290</CorrelationID>
<Version>553</Version>
<Build>e553_core_Bundled_6057805_R1</Build>
<NotificationEventName>FeedbackReceived</NotificationEventName>
<RecipientUserID>Seller1</RecipientUserID>
<FeedbackDetailArray>
<FeedbackDetail>
<CommentingUser>Buyer1</CommentingUser>
<CommentingUserScore>100</CommentingUserScore>
<CommentText>Thanks for the payment!</CommentText>
<CommentTime>2008-02-13T02:27:16.000Z</CommentTime>
<CommentType>Positive</CommentType>
<ItemID>250000627045</ItemID>
<Role>Buyer</Role>
<FeedbackID>11830457019</FeedbackID>
</FeedbackDetail>
</FeedbackDetailArray>
<FeedbackDetailItemTotal>215</FeedbackDetailItemTotal>
<FeedbackScore>1</FeedbackScore>
</GetFeedbackResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 59

FeedbackStarChanged

eBay sends a **FeedbackStarChanged** notification to third parties subscribed on the member's behalf whenever he attains a new Feedback Star level. The typical flow of events is:

- A third party subscribes to the **FeedbackStarChanged** notification on behalf of a member (buyer or seller).
- The buyer receives feedback that bumps him up to the next Feedback Star level.
- eBay sends the **FeedbackStarChanged** notification to the third party, indicating the member reached a new Feedback Star level.

eBay also sends **FeedbackStarChanged** notifications to the third party for all new Feedback Star levels a member attains.

Data Fields Returned

The data fields the **FeedbackStarChanged** notification returns are the same as those returned by **GetFeedbackResponse**. Check **GetFeedbackResponse** in the generated WsdlDoc documentation.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 59-1 Sample FeedbackStarChanged Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
  xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
  <ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">roYc91qpEAKq0mTvMoqSEg==</
  ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
< GetUserResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2008-02-14T23:46:49.253Z</Timestamp>
  <Ack>Success</Ack>
  <CorrelationID>137693230</CorrelationID>
  <Version>553</Version>
  <Build>e553_core_Bundled_6078830_R1</Build>
  <NotificationEventName>FeedbackStarChanged</NotificationEventName>
  <RecipientUserID>User1</RecipientUserID>
<User>
  <AboutMePage>false</AboutMePage>
  <EIAToken>nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wJk4CiCZmGpQWdj6x9nY+seQ==</EIAToken>
  <Email>...userid email goes here...</Email>
  <FeedbackScore>10</FeedbackScore>
  <UniqueNegativeFeedbackCount>0</UniqueNegativeFeedbackCount>
  <UniquePositiveFeedbackCount>10</UniquePositiveFeedbackCount>
  <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Yellow</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2007-01-09T17:51:56.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>User1</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2007-01-09T17:51:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <MerchandizingPref>OptIn</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerGuaranteeLevel>Not Eligible</SellerGuaranteeLevel>
    <SellerLevel>None</SellerLevel>
    <SchedulingInfo>
      <MaxScheduledMinutes>30240</MaxScheduledMinutes>
      <MinScheduledMinutes>0</MinScheduledMinutes>
      <MaxScheduledItems>1000</MaxScheduledItems>
    </SchedulingInfo>
    <StoreOwner>false</StoreOwner>
    <ExpressEligible>false</ExpressEligible>
    <PaymentMethod>CreditCard</PaymentMethod>
    <ExpressWallet>false</ExpressWallet>
    <CharityRegistered>false</CharityRegistered>
    <SafePaymentExempt>true</SafePaymentExempt>
  </SellerInfo>
  <eBayWikiReadOnly>false</eBayWikiReadOnly>
  <MotorsDealer>false</MotorsDealer>
  <UniqueNeutralFeedbackCount>0</UniqueNeutralFeedbackCount>
</User>
</ GetUserResponse>
</soapenv:Body>
</soapenv:Envelope>
```


Chapter 60

FixedPriceTransaction

A [FixedPriceTransaction](#) notification is sent to the seller when a listing for a Fixed Price item ends. A Fixed Price item ends when one of the following two conditions is met:

- Checkout is turned on, and the buyer completes the first step in the checkout flow by requesting a total.
- Checkout is turned off, and the buyer purchases the item.

Data Fields Returned

The [FixedPriceTransaction](#) notification returns the same data as the [GetItemTransactions](#) call with the ReturnAll detail level.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Chapter 61

INR (ItemNotReceived) Notifications

- “[INRBuyerOpenedDispute](#)” on page 685
- “[INRBuyerRespondedToDispute](#)” on page 685
- “[INRBuyerClosedDispute](#)” on page 686
- “[INRSellerRespondedToDispute](#)” on page 686

With **ItemNotReceived** notifications, buyers and sellers can be notified when an ItemNotReceived (INR) dispute that they are a part of is created or changes state. You will be notified when the following events occur:

- An INR is filed
- The buyer or seller involved in the INR dispute communicates with the other party (see “[AddDisputeResponse](#)” on page 775)
- The INR dispute is closed (see “[AddDisputeResponse](#)” on page 775)

Before using this notification, be sure that you have read and understood the information in “[Working with Platform Notifications](#)” on page 623.

INRBuyerOpenedDispute

Data Fields Returned

The data fields the **INRBuyerOpenedDispute** notification returns are the same as those returned by **GetDispute**. See **GetDisputeResponseType** in the Type documentation.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

INRBuyerRespondedToDispute

Data Fields Returned

The data fields the **INRBuyerRespondedToDispute** notification returns are the same as those returned by **GetDispute**. See **GetDisputeResponseType** in the Type documentation.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

INRBViewerClosedDispute

Data Fields Returned

The data fields the **INRBViewerClosedDispute** notification returns are the same as those returned by **GetDispute**. See **GetDisputeResponseType** in the Type documentation.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

INRSellerRespondedToDispute

Data Fields Returned

The data fields the **INRSellerRespondedToDispute** notification returns are the same as those returned by **GetDispute**. See **GetDisputeResponseType** in the Type documentation.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

The following sample shows a **INRSellerRespondedToDispute** notification. See “[GetDispute](#)” on page 828 for additional **GetDispute** output examples.

Example 61-1 Sample INRSellerRespondedToDispute Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents" xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">abc123==</ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetDisputeResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2005-08-19T06:00:17.146Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>95226300</CorrelationID>
<Version>421</Version>
<Build>e421_core_api5_1642313_R1</Build>
<NotificationEventName>INRSellerRespondedToDispute</NotificationEventName>
<RecipientUserID>someBuyer</RecipientUserID>
```

Example 61-1 Sample INRSellerRespondedToDispute Notification

```
<Dispute>
  <DisputeID>654321</DisputeID>
  <DisputeRecordType>ItemNotReceived</DisputeRecordType>
  <DisputeState>NotReceivedMutualCommunication</DisputeState>
  <DisputeStatus>WaitingForBuyerResponse</DisputeStatus>
  <BuyerUserID>someBuyer</BuyerUserID>
  <SellerUserID>someSeller</SellerUserID>
  <TransactionID>0</TransactionID>
  <Item>
    <ItemID>123456789</ItemID>
    <ListingDetails>
      <StartTime>2005-08-09T03:55:13.000Z</StartTime>
      <EndTime>2005-08-09T03:55:15.000Z</EndTime>
    </ListingDetails>
    <Quantity>1</Quantity>
    <SellingStatus>
      <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
      <CurrentPrice currencyID="USD">1.0</CurrentPrice>
    </SellingStatus>
    <Site>US</Site>
    <Title>INR notification</Title>
  </Item>
  <disputeReason>ItemNotReceived</DisputeReason>
  <DisputeExplanation>Unspecified</DisputeExplanation>
  <DisputeCreditEligibility>InEligible</DisputeCreditEligibility>
  <DisputeCreatedTime>2005-08-19T04:23:22.000Z</DisputeCreatedTime>
  <DisputeModifiedTime>2005-08-19T06:00:15.000Z</DisputeModifiedTime>
  <DisputeMessage>
    <MessageID>123456</MessageID>
    <MessageSource>Buyer</MessageSource>
    <MessageCreationTime>2005-08-19T04:23:24.000Z</MessageCreationTime>
    <MessageText>Item Not Received: INR SR2D notification (#123456789)
Payment Method: Personal Check
Payment Date: Aug-08-2005</MessageText>
  </DisputeMessage>
  <DisputeMessage>
    <MessageID>234567</MessageID>
    <MessageSource>Seller</MessageSource>
    <MessageCreationTime>2005-08-19T05:58:07.000Z</MessageCreationTime>
    <MessageText>Where's my stuff?</MessageText>
  </DisputeMessage>
  <DisputeMessage>
    <MessageID>345678</MessageID>
    <MessageSource>Seller</MessageSource>
    <MessageCreationTime>2005-08-19T06:00:15.000Z</MessageCreationTime>
    <MessageText>Where's my stuff?</MessageText>
  </DisputeMessage>
  <Escalation>true</Escalation>
  <PurchaseProtection>false</PurchaseProtection>
</Dispute>
</GetDisputeResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 62

ItemAddedToBidGroup

An [ItemAddedToBidGroup](#) notification is sent to a subscribing user when a buyer places an item in his or her bid group, as discussed in [Managing Buying and Watch Lists with My eBay](#).

Data Fields Returned

The [ItemAddedToBidGroup](#) notification returns the same data as [GetItem Call](#) with the default detail level.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

Example 62-1 Sample ItemAddedToBidGroup Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">KyROpBcPQJIFRbSHDBlnCw==</
ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-02-21T02:18:29.703Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>137932150</CorrelationID>
<Version>553</Version>
<Build>e553_core_Bundled_6114677_R1</Build>
<NotificationEventName>ItemAddedToBidGroup</NotificationEventName>
<RecipientUserID>Buyer1</RecipientUserID>

<Item>
<AutoPay>false</AutoPay>
<BuyerProtection>ItemEligible</BuyerProtection>
<BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
<Country>US</Country>
<Currency>USD</Currency>
<GiftIcon>0</GiftIcon>
<HitCounter>NoHitCounter</HitCounter>
<ItemID>250000632012</ItemID>
<ListingDetails>
<Adult>false</Adult>
<BindingAuction>false</BindingAuction>
<CheckoutEnabled>true</CheckoutEnabled>
<ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
<ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
<HasReservePrice>false</HasReservePrice>
<StartTime>2008-02-17T01:41:10.000Z</StartTime>
<EndTime>2008-02-22T01:41:10.000Z</EndTime>
<ViewItemURL>http://cgi.qa-api011.qa.ebay.com/ws/
eBayISAPI.dll?ViewItem&item=250000632012&category=1463</ViewItemURL>
<HasUnansweredQuestions>false</HasUnansweredQuestions>
<HasPublicMessages>false</HasPublicMessages>
<BuyItNowAvailable>true</BuyItNowAvailable>
<ExpressListing>false</ExpressListing>
<ViewItemURLForNaturalSearch>http://cgi.qa-api011.qa.ebay.com/Test-Item-AddedTo-RemovedFrom-
BidGroup-1_W0QQitemZ250000632012QQcategoryZ1463QQcmdZViewItem</ViewItemURLForNaturalSearch>
</ListingDetails>
<ListingDuration>Days_5</ListingDuration>
<ListingType>Chinese</ListingType>
<Location>San Jose, CA</Location>
<PaymentMethods>PaymentSeeDescription</PaymentMethods>
<PaymentMethods>PayPal</PaymentMethods>
<PrimaryCategory>
<CategoryID>1463</CategoryID>
<CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
</PrimaryCategory>
<PrivateListing>false</PrivateListing>
<Quantity>1</Quantity>
<ReviseStatus>
<ItemRevised>false</ItemRevised>
</ReviseStatus>
```

Example 62-1 Sample ItemAddedToBidGroup Notification (Continued)

```

<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>Invalid Request</Email>
  <FeedbackScore>100</FeedbackScore>
  <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>...seller userid goes here...</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <MerchandizingPref>Opt In</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerLevel>None</SellerLevel>
    <StoreOwner>false</StoreOwner>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
    <SafePaymentExempt>true</SafePaymentExempt>
  </SellerInfo>
  <MotorsDealer>false</MotorsDealer>
</Seller>

<SellingStatus>
  <BidCount>0</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.0</CurrentPrice>
  <MinimumToBid currencyID="USD">1.0</MinimumToBid>
  <QuantitySold>0</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>false</SecondChanceEligible>
  <ListingStatus>Active</ListingStatus>
</SellingStatus>

<ShippingDetails>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
</ShippingDetails>

<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>PT23H22M41S</TimeLeft>
<Title>Test Item{AddedTo,RemovedFrom}BidGroup 1</Title>
<HitCount>1</HitCount>
<GetItFast>false</GetItFast>
<PostalCode></PostalCode>
<PictureDetails>
  <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>

```

Chapter 63

ItemAddedToWatchList

eBay sends an **ItemAddedToWatchList** notification to third parties subscribed on the buyer's behalf whenever a buyer adds an item to his watch list. The typical flow of events is:

- A third party subscribes to the **ItemAddedToWatchList** notification on behalf of a buyer.
- The buyer adds an item to his or her watch list.
- eBay sends the **ItemAddedToWatchList** notification to the third party, indicating the item was just added to the watch list.

eBay also sends **ItemAddedToWatchList** notifications to the third party for all items the buyer adds, either through the eBay sites or programmatically from an API or SDK.

Data Fields Returned

The data fields the **ItemAddedToWatchList** notification returns are the same as those returned by **GetItemResponse** with the default detail level.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

Example 63-1 Sample ItemAddedToWatchList Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebc:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
  xmlns:ebc="urn:ebay:apis:eBLBaseComponents">
<ebc:NotificationSignature xmlns:ebc="urn:ebay:apis:eBLBaseComponents">wÜyT2INFKPU7Y1e9vKjmQA==</
  ebc:NotificationSignature>
</ebc:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
  <GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2008-02-12T00:07:45.500Z</Timestamp>
    <Ack>Success</Ack>
    <CorrelationID>137519850</CorrelationID>
    <Version>553</Version>
    <Build>e553_core_Bundled_6057805_R1</Build>
    <NotificationEventName>ItemAddedToWatchList</NotificationEventName>
    <RecipientUserID>Buyer1</RecipientUserID>
```

Example 63-1 Sample ItemAddedToWatchList Notification

```
<Item>
  <AutoPay>false</AutoPay>
  <BuyerProtection>ItemEligible</BuyerProtection>
  <BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
  <Country>US</Country>
  <Currency>USD</Currency>
  <GiftIcon>0</GiftIcon>
  <HitCounter>NoHitCounter</HitCounter>
  <ItemID>250000626026</ItemID>
  <ListingDetails>
    <Adult>false</Adult>
    <BindingAuction>false</BindingAuction>
    <CheckoutEnabled>true</CheckoutEnabled>
    <ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
    <ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
    <HasReservePrice>false</HasReservePrice>
    <StartTime>2008-02-12T00:06:25.000Z</StartTime>
    <EndTime>2008-02-17T00:06:25.000Z</EndTime>
    <ViewItemURL>http://cgi.qa-api012.qa.ebay.com/ws/
eBayISAPI.dll?ViewItem&item=250000626026&category=1463</ViewItemURL>
    <HasUnansweredQuestions>false</HasUnansweredQuestions>
    <HasPublicMessages>false</HasPublicMessages>
    <BuyItNowAvailable>true</BuyItNowAvailable>
    <ExpressListing>false</ExpressListing>
    <ViewItemURLForNaturalSearch>http://cgi.ebay.com/Test-ItemAddedToWatchList-
notify_W0QitemZ250000626026QQcategoryZ1463QcmdZViewItem</ViewItemURLForNaturalSearch>
  </ListingDetails>
  <ListingDuration>Days_5</ListingDuration>
  <ListingType>Chinese</ListingType>
  <Location>San Jose, CA</Location>
  <PaymentMethods>PaymentSeeDescription</PaymentMethods>
  <PaymentMethods>PayPal</PaymentMethods>
  <PrimaryCategory>
    <CategoryID>1463</CategoryID>
    <CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
  </PrimaryCategory>
  <PrivateListing>false</PrivateListing>
  <Quantity>1</Quantity>
  <ReviseStatus>
    <ItemRevised>false</ItemRevised>
  </ReviseStatus>
  <Seller>
    <AboutMePage>false</AboutMePage>
    <Email>Invalid Request</Email>
    <FeedbackScore>100</FeedbackScore>
    <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
    <FeedbackPrivate>false</FeedbackPrivate>
    <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
    <IDVerified>false</IDVerified>
    <eBayGoodStanding>true</eBayGoodStanding>
    <NewUser>false</NewUser>
    <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
    <Site>US</Site>
    <Status>Confirmed</Status>
    <UserID>Seller1</UserID>
    <UserIDChanged>false</UserIDChanged>
    <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
    <VATStatus>NoVATTax</VATStatus>
    <SellerInfo>
      <AllowPaymentEdit>true</AllowPaymentEdit>
      <CheckoutEnabled>true</CheckoutEnabled>
      <CIPBankAccountStored>false</CIPBankAccountStored>
      <GoodStanding>true</GoodStanding>
      <MerchandizingPref>OptIn</MerchandizingPref>
      <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
      <SellerLevel>None</SellerLevel>
      <StoreOwner>false</StoreOwner>
      <ExpressEligible>false</ExpressEligible>
      <ExpressWallet>false</ExpressWallet>
      <SafePaymentExempt>true</SafePaymentExempt>
    </SellerInfo>
    <MotorsDealer>false</MotorsDealer>
  </Seller>
```

Example 63-1 Sample ItemAddedToWatchList Notification

```
<SellingStatus>
    <BidCount>0</BidCount>
    <BidIncrement currencyID="USD">0.25</BidIncrement>
    <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
    <CurrentPrice currencyID="USD">1.0</CurrentPrice>
    <MinimumToBid currencyID="USD">1.0</MinimumToBid>
    <QuantitySold>0</QuantitySold>
    <ReserveMet>true</ReserveMet>
    <SecondChanceEligible>false</SecondChanceEligible>
    <ListingStatus>Active</ListingStatus>
</SellingStatus>

<ShippingDetails>
    <ApplyShippingDiscount>false</ApplyShippingDiscount>
    <InsuranceOption>NotOffered</InsuranceOption>
    <SalesTax>
        <SalesTaxPercent>0.0</SalesTaxPercent>
        <ShippingIncludedInTax>false</ShippingIncludedInTax>
    </SalesTax>
    <ThirdPartyCheckout>false</ThirdPartyCheckout>
    <TaxTable/>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>P4DT23H58M40S</TimeLeft>
<Title>Test ItemAddedToWatchList notify</Title>
<HitCount>0</HitCount>
<GetItFast>false</GetItFast>
<PostalCode>
</PostalCode>
<PictureDetails>
    <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 64

ItemClosed

eBay sends an **ItemClosed** notification to third parties subscribed on behalf of a buyer or seller when a listing closes. The notification is triggered by **ItemWon**, **ItemSold**, and **ItemUnsold** events. The typical flow of events is:

- A third party subscribes to the **ItemClosed** notification on behalf of a buyer or seller.
- The item closes with an **ItemSold**, an **ItemUnsold**, or an **ItemWon** event.
- eBay sends the **ItemClosed** notification to the third party, indicating the item has closed.

Data Fields Returned

The data fields the **ItemClosed** notification returns are the same as those returned by **GetItemResponse** with the default detail level.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

Example 64-1 Sample ItemClosed Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">FvGSpWlz6e2lggkXkig4Nw==</
ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
```

Example 64-1 Sample ItemClosed Notification

```
<soapenv:Body>
  <GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2010-05-10T21:59:36.810Z</Timestamp>
    <Ack>Success</Ack>
    <CorrelationID>205564160</CorrelationID>
    <Version>669</Version>
    <Build>E669_CORE_BUNDLED_11196875_R1</Build>
    <NotificationEventName>ItemClosed</NotificationEventName>
    <RecipientUserID>seller1_ebay</RecipientUserID>
    <EIAToken>nY+sHZ2PrBmdj6wNnY+sEZ2PrA2dj6wJkoejAJKFqQqdj6x9nY+seQ**</EIAToken>
    <Item>
      <AutoPay>false</AutoPay>
      <BuyerProtection>ItemEligible</BuyerProtection>
      <BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
      <Country>US</Country>
      <Currency>USD</Currency>
      <GiftIcon>0</GiftIcon>
      <HitCounter>NoHitCounter</HitCounter>
      <ItemID>250001445529</ItemID>
      <ListingDetails>
        <Adult>false</Adult>
        <BindingAuction>false</BindingAuction>
        <CheckoutEnabled>true</CheckoutEnabled>
        <ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
        <ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
        <ConvertedReservePrice currencyID="USD">0.0</ConvertedReservePrice>
        <HasReservePrice>false</HasReservePrice>
        <StartTime>2010-05-10T21:30:45.000Z</StartTime>
        <EndTime>2010-05-10T21:31:26.000Z</EndTime>
        <ViewItemURL>http://cgi.qa.ebay.com/ws/
          eBayISAPI.dll?ViewItem&item=250001445529&category=357</ViewItemURL>
        <HasUnansweredQuestions>false</HasUnansweredQuestions>
        <HasPublicMessages>false</HasPublicMessages>
        <BuyItNowAvailable>true</BuyItNowAvailable>
        <ExpressListing>false</ExpressListing>
        <ExpressItemRequirements/>
        <ViewItemURLForNaturalSearch>http://cgi.qa.ebay.com/Test-Norman-ItemUnsold-
          notify_W0QQitemZ250001445529QQcategoryZ357QQcmdZViewItem</ViewItemURLForNaturalSearch>
      </ListingDetails>
      <ListingDuration>Days_1</ListingDuration>
      <ListingType>Chinese</ListingType>
      <Location>San Jose, CA</Location>
      <PaymentMethods>PayPal</PaymentMethods>
      <PayPalEmailAddress>ve2@aol.com</PayPalEmailAddress>
      <PrimaryCategory>
        <CategoryID>357</CategoryID>
        <CategoryName>Art:Art from Dealers & Resellers:Folk Art & Primitives</CategoryName>
      </PrimaryCategory>
      <PrivateListing>false</PrivateListing>
      <Quantity>1</Quantity>
      <ReservePrice currencyID="USD">0.0</ReservePrice>
      <ReviseStatus>
        <ItemRevised>false</ItemRevised>
      </ReviseStatus>
      <Seller>
        <AboutMePage>false</AboutMePage>
        <Email>nc_seller1@unicorn.qa.ebay.com</Email>
        <FeedbackScore>100</FeedbackScore>
        <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
        <FeedbackPrivate>false</FeedbackPrivate>
        <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
        <IDVerified>false</IDVerified>
        <eBayGoodStanding>true</eBayGoodStanding>
        <NewUser>false</NewUser>
        <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
        <Site>US</Site>
        <Status>Confirmed</Status>
        <UserID>seller1_ebay</UserID>
        <UserIDChanged>false</UserIDChanged>
        <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
        <VATStatus>NoVATTax</VATStatus>
      </Seller>
    </Item>
  </GetItemResponse>
</soapenv:Body>
```

Example 64-1 Sample ItemClosed Notification

```
<SellerInfo>
  <AllowPaymentEdit>true</AllowPaymentEdit>
  <CheckoutEnabled>true</CheckoutEnabled>
  <CIBBankAccountStored>false</CIBBankAccountStored>
  <GoodStanding>true</GoodStanding>
  <LiveAuctionAuthorized>false</LiveAuctionAuthorized>
  <MerchandizingPref>OptIn</MerchandizingPref>
  <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
  <SellerLevel>None</SellerLevel>
  <StoreOwner>false</StoreOwner>
  <ExpressEligible>false</ExpressEligible>
  <ExpressWallet>false</ExpressWallet>
  <SafePaymentExempt>true</SafePaymentExempt>
</SellerInfo>
<MotorsDealer>false</MotorsDealer>
</Seller>
<SellingStatus>
  <BidCount>0</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.0</CurrentPrice>
  <LeadCount>0</LeadCount>
  <MinimumToBid currencyID="USD">1.0</MinimumToBid>
  <QuantitySold></QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>false</SecondChanceEligible>
  <ListingStatus>Completed</ListingStatus>
</SellingStatus>
<ShippingDetails>
  <AllowPaymentEdit>true</AllowPaymentEdit>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceFee currencyID="USD">0.0</InsuranceFee>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ShippingServiceOptions>
    <ShippingService>UPSGround</ShippingService>
    <ShippingServiceCost currencyID="USD">5.0</ShippingServiceCost>
    <ShippingServicePriority>1</ShippingServicePriority>
    <ExpeditedService>false</ExpeditedService>
    <ShippingTimeMin>1</ShippingTimeMin>
    <ShippingTimeMax>6</ShippingTimeMax>
  </ShippingServiceOptions>
  <ShippingType>Flat</ShippingType>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
  <InsuranceDetails>
    <InsuranceOption>NotOffered</InsuranceOption>
  </InsuranceDetails>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
  <StartPrice currencyID="USD">1.0</StartPrice>
  <SubTitle>sub title</SubTitle>
  <TimeLeft>PT0S</TimeLeft>
  <Title>Test Norman ItemUnsold notify</Title>
  <HitCount>0</HitCount>
  <GetItFast>false</GetItFast>
  <PostalCode></PostalCode>
  <PictureDetails>
    <PhotoDisplay>None</PhotoDisplay>
  </PictureDetails>
  <DispatchTimeMax>5</DispatchTimeMax>
  <ProxyItem>false</ProxyItem>
  <BuyerGuaranteePrice currencyID="USD">20.0</BuyerGuaranteePrice>
  <IntangibleItem>false</IntangibleItem>
```

Example 64-1 Sample ItemClosed Notification

```
<ReturnPolicy>
  <RefundOption>Exchange</RefundOption>
  <Refund>Exchange</Refund>
  <>ReturnsWithinOption>Days_30</ReturnsWithinOption>
  <ReturnsWithin>30 Days</ReturnsWithin>
  <>ReturnsAcceptedOption>ReturnsAccepted</>ReturnsAcceptedOption>
  <>ReturnsAccepted>Returns Accepted</>ReturnsAccepted>
  <Description>in here . . , :)</Description>
  <ShippingCostPaidByOption>Buyer</ShippingCostPaidByOption>
  <ShippingCostPaidBy>Buyer</ShippingCostPaidBy>
</ReturnPolicy>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 65

ItemListed

eBay sends an **ItemListed** notification to third parties subscribed on the seller's behalf whenever a seller lists or relists an item. The typical flow of events is:

- A third party subscribes to the **ItemListed** notification on behalf of a seller.
- The seller lists or relists an item.
- eBay sends the **ItemListed** notification to the third party, indicating the item was just listed.
- eBay also sends **ItemListed** notifications to the third party for all items the seller lists, either through the eBay sites or programmatically from an API or SDK.

The maximum amount of **ItemListed** notifications eBay can send is 1 for each item a subscribed application is monitoring.

Data Fields Returned

The data fields the **ItemListed** notification returns are the same as those returned by **GetItem** with the default detail level. Check **GetItemResponse** (which contains **Item**) in the generated WSDLDoc documentation.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

In this example, seller MagicalBookSeller is notified that her item, a copy of "Harry Potter and the Philosopher's Stone," has successfully been listed on eBay.

Example 65-1 Sample ItemListed Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
  <ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents" xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
    <ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">DYrq65GiZvg4cLFFW8J/Mg==</ebl:NotificationSignature>
    </ebl:RequesterCredentials>
  </soapenv:Header>
<soapenv:Body>
  <GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2011-01-06T08:08:13.025Z</Timestamp>
    <Ack>Success</Ack>
    <CorrelationID>476457080</CorrelationID>
    <Version>699</Version>
    <Build>E699_CORE_BUNDLED_12457306_R1</Build>
    <NotificationEventName>ItemListed</NotificationEventName>
    <RecipientUserID>testuser_magicalbookseller</RecipientUserID>
    <EIAToken>nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wFk4CpDZSLpwqdj6x9nY+seQ**</EIAToken>
    <Item>
      <AutoPay>false</AutoPay>
      <BuyerProtection>ItemIneligible</BuyerProtection>
      <BuyItNowPrice currencyID="USD">0.0</BuyItNowPrice>
      <Country>US</Country>
      <Currency>USD</Currency>
      <GiftIcon>0</GiftIcon>
      <HitCounter>NoHitCounter</HitCounter>
      <ItemID>110066229557</ItemID>
      <ListingDetails>
        <Adult>false</Adult>
        <BindingAuction>false</BindingAuction>
        <CheckoutEnabled>true</CheckoutEnabled>
        <ConvertedBuyItNowPrice currencyID="USD">0.0</ConvertedBuyItNowPrice>
        <ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
        <ConvertedReservePrice currencyID="USD">0.0</ConvertedReservePrice>
        <HasReservePrice>false</HasReservePrice>
        <StartTime>2011-01-06T08:07:54.000Z</StartTime>
        <EndTime>2011-01-13T08:07:54.000Z</EndTime>
        <ViewItemURL>http://cgi.sandbox.ebay.com/ws/eBayISAPI.dll?ViewItem&item=110066229557&category=377</ViewItemURL>
        <HasUnansweredQuestions>false</HasUnansweredQuestions>
        <HasPublicMessages>false</HasPublicMessages>
        <ExpressListing>false</ExpressListing>
      </ListingDetails>
      <ListingDuration>Days_7</ListingDuration>
      <ListingType>Chinese</ListingType>
      <Location>San Jose, California</Location>
      <PaymentMethods>PayPal</PaymentMethods>
      <PayPalEmailAddress>magicalbookseller@yahoo.com</PayPalEmailAddress>
      <PrimaryCategory>
        <CategoryID>377</CategoryID>
        <CategoryName>Books:Fiction & Literature</CategoryName>
      </PrimaryCategory>
      <PrivateListing>false</PrivateListing>
      <Quantity>1</Quantity>
      <ReservePrice currencyID="USD">0.0</ReservePrice>
      <ReviseStatus>
        <ItemRevised>false</ItemRevised>
      </ReviseStatus>
    </Item>
  </GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Example 65-1 Sample ItemListed Notification

```
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>testuser_magicalbookseller@yahoo.com</Email>
  <FeedbackScore>31</FeedbackScore>
  <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Yellow</FeedbackRatingStar>
  <IDVerified>true</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2010-01-01T00:00:00.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>testuser_magicalbookseller</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2007-11-29T18:43:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <LiveAuctionAuthorized>false</LiveAuctionAuthorized>
    <MerchandizingPref>OptIn</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <StoreOwner>true</StoreOwner>
    <StoreURL>http://www.stores.sandbox.ebay.com/id=132854966</StoreURL>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
    <SafePaymentExempt>false</SafePaymentExempt>
  </SellerInfo>
  <MotorsDealer>false</MotorsDealer>
</Seller>

<SellingStatus>
  <BidCount>0</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.0</CurrentPrice>
  <LeadCount>0</LeadCount>
  <MinimumToBid currencyID="USD">1.0</MinimumToBid>
  <QuantitySold>0</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>false</SecondChanceEligible>
  <ListingStatus>Active</ListingStatus>
</SellingStatus>

<ShippingDetails>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <CalculatedShippingRate>
    <OriginatingPostalCode>95125</OriginatingPostalCode>
    <PackageDepth measurementSystem="English" unit="inches">2</PackageDepth>
    <PackageLength measurementSystem="English" unit="inches">10</PackageLength>
    <PackageWidth measurementSystem="English" unit="inches">7</PackageWidth>
    <PackagingHandlingCosts currencyID="USD">0.0</PackagingHandlingCosts>
    <ShippingIrregular>false</ShippingIrregular>
    <ShippingPackage>PackageThickEnvelope</ShippingPackage>
    <WeightMajor measurementSystem="English" unit="lbs">2</WeightMajor>
    <WeightMinor measurementSystem="English" unit="oz">0</WeightMinor>
  </CalculatedShippingRate>
</SalesTax>
  <SalesTaxPercent>0.0</SalesTaxPercent>
  <ShippingIncludedInTax>false</ShippingIncludedInTax>
</SalesTax>
  <ShippingServiceOptions>
    <ShippingService>USPSMedia</ShippingService>
    <ShippingServicePriority>1</ShippingServicePriority>
    <ExpeditedService>false</ExpeditedService>
    <ShippingTimeMin>2</ShippingTimeMin>
    <ShippingTimeMax>9</ShippingTimeMax>
  </ShippingServiceOptions>
```

Example 65-1 Sample ItemListed Notification

```
<ShippingType>Calculated</ShippingType>
<ThirdPartyCheckout>false</ThirdPartyCheckout>
<ExcludeShipToLocation>Asia</ExcludeShipToLocation>
<ExcludeShipToLocation>Middle East</ExcludeShipToLocation>
<ExcludeShipToLocation>Southeast Asia</ExcludeShipToLocation>
<SellerExcludeShipToLocationsPreference>true</SellerExcludeShipToLocationsPreference>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<Storefront>
<StoreCategoryID>1</StoreCategoryID>
<StoreCategory2ID>0</StoreCategory2ID>
<StoreURL>http://www.stores.sandbox.ebay.com/id=132854966</StoreURL>
</Storefront>
<TimeLeft>P6DT23H59M42S</TimeLeft>
<Title>Harry Potter and the Philosopher's Stone</Title>
<UUID>CD751A9973E249ABB880E1B2DF65B50B</UUID>
<HitCount>0</HitCount>
<LocationDefaulted>true</LocationDefaulted>
<GetItFast>false</GetItFast>
<PostalCode>95125</PostalCode>
<PictureDetails>
<GalleryType>Gallery</GalleryType>
<GalleryURL>http://il.sandbox.ebayimg.com/03/i/00/3e/60/d7_1.JPG?set_id=8800005007</GalleryURL>
<PhotoDisplay>PicturePack</PhotoDisplay>
<PictureURL>http://il.sandbox.ebayimg.com/03/i/00/3e/60/d7_1.JPG?set_id=8800005007</PictureURL>
</PictureDetails>
<DispatchTimeMax>3</DispatchTimeMax>
<ProxyItem>false</ProxyItem>
<BuyerGuaranteePrice currencyID="USD">20000.0</BuyerGuaranteePrice>
<IntangibleItem>false</IntangibleItem>
<ReturnPolicy>
<RefundOption>MoneyBack</RefundOption>
<Refund>Money Back</Refund>
<ReturnsWithinOption>Days_30</ReturnsWithinOption>
<ReturnsWithin>30 Days</ReturnsWithin>
<ReturnsAcceptedOption>ReturnsAccepted</ReturnsAcceptedOption>
<ReturnsAccepted>Returns Accepted</ReturnsAccepted>
<Description>This is the first book in the Harry Potter series. In excellent condition!</Description>
<ShippingCostPaidByOption>Buyer</ShippingCostPaidByOption>
<ShippingCostPaidBy>Buyer</ShippingCostPaidBy>
</ReturnPolicy>
<PaymentAllowedSite>eBayMotors</PaymentAllowedSite>
<PaymentAllowedSite>CanadaFrench</PaymentAllowedSite>
<PaymentAllowedSite>Canada</PaymentAllowedSite>
<PaymentAllowedSite>US</PaymentAllowedSite>
<ConditionID>1000</ConditionID>
<ConditionDisplayName>Brand New</ConditionDisplayName>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 66

ItemLost

eBay sends an **ItemLost** notification to third parties subscribed on the buyer's behalf whenever a buyer loses an item sold by auction. This notification is triggered by the end of the auction, not just when a user has been outbid. This notification needs to support items that the buyer attempted to purchase using Best Offer with no modification of the payload for Client Alerts. Note: This notification also handles the loss of auction items that include a Buy It Now option. The typical flow of events is:

- A third party subscribes to the **ItemLost** notification on behalf of a buyer.
- The buyer loses an item by not having the highest bid in an auction or by placing an unsuccessful Best Offer.
- eBay sends the **ItemLost** notification to the third party, indicating the user just lost an item, either by auction or by Best Offer.

eBay also sends **ItemLost** notifications to the third party for all items the buyer loses.

Data Fields Returned

The data fields the **ItemLost** notification returns are the same as those returned by **GetItemResponse** with the default detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 66-1 Sample ItemLost Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">FT0y5Jkxy9gPJgZqQzTEiQ==</
ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-02-13T04:29:46.590Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>137541650</CorrelationID>
<Version>553</Version>
<Build>e553_core_Bundled_6057805_R1</Build>
<NotificationEventName>ItemLost</NotificationEventName>
<RecipientUserID>Buyer1</RecipientUserID>
<Item>
<AutoPay>false</AutoPay>
<BuyerProtection>ItemEligible</BuyerProtection>
<BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
<Country>US</Country>
<Currency>USD</Currency>
<GiftIcon>0</GiftIcon>
<HitCounter>NoHitCounter</HitCounter>
<ItemID>250000627138</ItemID>
<ListingDetails>
<Adult>false</Adult>
<BindingAuction>false</BindingAuction>
<CheckoutEnabled>true</CheckoutEnabled>
<ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
<ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
<HasReservePrice>false</HasReservePrice>
<StartTime>2008-02-13T04:27:07.000Z</StartTime>
<EndTime>2008-02-13T04:27:56.000Z</EndTime>
<ViewItemURL>http://cgi.qa-api012.qa.ebay.com/ws/
eBayISAPI.dll?ViewItem&amp;item=250000627138&amp;category=1463</ViewItemURL>
<HasUnansweredQuestions>false</HasUnansweredQuestions>
<HasPublicMessages>false</HasPublicMessages>
<ExpressListing>false</ExpressListing>
<ViewItemURLForNaturalSearch>http://cgi.qa-api012.qa.ebay.com/Test-ItemLost-
notify_W0QQitemZ250000627138QQcategoryZ1463QQcmdZViewItem</ViewItemURLForNaturalSearch>
</ListingDetails>
```

Example 66-1 Sample ItemLost Notification

```
<ListingDuration>Days_5</ListingDuration>
<ListingType>Chinese</ListingType>
<Location>San Jose, CA</Location>
<PaymentMethods>PaymentSeeDescription</PaymentMethods>
<PaymentMethods>PayPal</PaymentMethods>
<PrimaryCategory>
  <CategoryID>1463</CategoryID>
  <CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
</PrimaryCategory>
<PrivateListing>false</PrivateListing>
<Quantity>1</Quantity>
<RevisedStatus>
  <ItemRevised>false</ItemRevised>
</RevisedStatus>
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>Invalid Request</Email>
  <FeedbackScore>100</FeedbackScore>
  <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>Seller1</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <MerchandizingPref>OptIn</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerLevel>None</SellerLevel>
    <StoreOwner>false</StoreOwner>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
    <SafePaymentExempt>true</SafePaymentExempt>
  </SellerInfo>
  <MotorsDealer>false</MotorsDealer>
</Seller>
```

Example 66-1 Sample ItemLost Notification

```
<SellingStatus>
  <BidCount>2</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.25</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.25</CurrentPrice>
  <HighBidder>
    <AboutMePage>false</AboutMePage>
    <EIASToken>nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wJkoWoCJaGoAmdj6x9nY+seQ==</EIASToken>
    <Email>Invalid Request</Email>
    <FeedbackScore>5</FeedbackScore>
    <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
    <FeedbackPrivate>false</FeedbackPrivate>
    <FeedbackRatingStar>None</FeedbackRatingStar>
    <IDVerified>false</IDVerified>
    <eBayGoodStanding>true</eBayGoodStanding>
    <NewUser>false</NewUser>
    <RegistrationDate>2004-02-24T07:00:00.000Z</RegistrationDate>
    <Site>US</Site>
    <Status>Confirmed</Status>
    <UserID>Buyer1</UserID>
    <UserIDChanged>false</UserIDChanged>
    <UserIDLastChanged>2004-02-24T07:00:00.000Z</UserIDLastChanged>
    <VATStatus>NoVATTax</VATStatus>
    <BuyerInfo>
      <ShippingAddress>
        <Country>US</Country>
        <PostalCode>95125</PostalCode>
      </ShippingAddress>
    </BuyerInfo>
    <UserAnonymized>false</UserAnonymized>
  </HighBidder>
  <MinimumToBid currencyID="USD">1.5</MinimumToBid>
  <QuantitySold>1</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>true</SecondChanceEligible>
  <ListingStatus>Completed</ListingStatus>
</SellingStatus>
<ShippingDetails>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>PT0S</TimeLeft>
<Title>Test ItemLost notify</Title>
<HitCount>0</HitCount>
<GetItFast>false</GetItFast>
<PostalCode>
</PostalCode>
<PictureDetails>
  <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 67

ItemMarkedPaid

eBay sends a **ItemMarkedPaid** notification to the seller whenever a seller who has sold an item from eBay, no matter it is an auction item, a fixed priced store item, a fixed priced item (BIN), or an auction item sold with BestOffer. The typical flow of events is:

A third party subscribes to the **ItemMarkedPaid** notification on behalf of a seller.

A buyer buys the item.

eBay sends the **ItemMarkedPaid** notification to the third party, indicating the seller sold the item before the listing expiration.

Data Fields Returned

The data fields the **ItemMarkedPaid** notification returns are the same as those returned by **GetItemResponse**. Check **GetItemResponse** in the generated WsdlDoc documentation.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

Example 67-1 Sample ItemMarkedPaid Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">Z2yhKdKmS0Ga5VPMLOAlg==</
ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-02-13T03:47:28.106Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>137541140</CorrelationID>
<Version>553</Version>
<Build>e553_core_Bundled_6057805_R1</Build>
<NotificationEventName>ItemMarkedPaid</NotificationEventName>
<RecipientUserID>Seller1</RecipientUserID>
<Item>
<AutoPay>false</AutoPay>
<BuyerProtection>ItemEligible</BuyerProtection>
<BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
<Country>US</Country>
<Currency>USD</Currency>
<GiftIcon>0</GiftIcon>
<HitCounter>NoHitCounter</HitCounter>
<ItemID>250000627102</ItemID>
<ListingDetails>
<Adult>false</Adult>
<BindingAuction>false</BindingAuction>
<CheckoutEnabled>true</CheckoutEnabled>
<ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
<ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
<ConvertedReservePrice currencyID="USD">0.0</ConvertedReservePrice>
<HasReservePrice>false</HasReservePrice>
<StartTime>2008-02-13T03:42:45.000Z</StartTime>
<EndTime>2008-02-13T03:43:32.000Z</EndTime>
<ViewItemURL>http://cgi.qa-api012.qa.ebay.com/ws/
eBayISAPI.dll?ViewItem&item=250000627102&category=1463</ViewItemURL>
<HasUnansweredQuestions>false</HasUnansweredQuestions>
<HasPublicMessages>false</HasPublicMessages>
<ExpressListing>false</ExpressListing>
<ViewItemURLForNaturalSearch>http://cgi.qa-api012.qa.ebay.com/Test-Item-Won-Sold-
notify_WQQitemZ250000627102QQcategoryZ1463QcmdZViewItem</ViewItemURLForNaturalSearch>
</ListingDetails>
<ListingDuration>Days_5</ListingDuration>
<ListingType>Chinese</ListingType>
<Location>San Jose, CA</Location>
<PaymentMethods>PaymentSeeDescription</PaymentMethods>
<PaymentMethods>PayPal</PaymentMethods>
<PayPalEmailAddress>ve2@aol.com</PayPalEmailAddress>
<PrimaryCategory>
<CategoryID>1463</CategoryID>
<CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
</PrimaryCategory>
<PrivateListing>false</PrivateListing>
<Quantity>1</Quantity>
<ReservePrice currencyID="USD">0.0</ReservePrice>
<ReviseStatus>
<ItemRevised>false</ItemRevised>
</ReviseStatus>
```

Example 67-1 Sample ItemMarkedPaid Notification

```
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>seller@email.com</Email>
  <FeedbackScore>100</FeedbackScore>
  <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>Seller1</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <LiveAuctionAuthorized>false</LiveAuctionAuthorized>
    <MerchandizingPref>OptIn</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerLevel>None</SellerLevel>
    <StoreOwner>false</StoreOwner>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
    <SafePaymentExempt>true</SafePaymentExempt>
  </SellerInfo>
  <MotorsDealer>false</MotorsDealer>
</Seller>
<SellingStatus>
  <BidCount>2</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.25</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.25</CurrentPrice>
  <HighBidder>
    <AboutMePage>false</AboutMePage>
    <EIASToken>nY+SHZ2PrBmdj6wVnY+sEZ2PrA2dj6wJkoWoCJaGoAmdj6x9nY+seQ==</EIASToken>
    <Email>buyer@email.com</Email>
    <FeedbackScore>5</FeedbackScore>
    <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
    <FeedbackPrivate>false</FeedbackPrivate>
    <FeedbackRatingStar>None</FeedbackRatingStar>
    <IDVerified>false</IDVerified>
    <eBayGoodStanding>true</eBayGoodStanding>
    <NewUser>false</NewUser>
    <RegistrationDate>2004-02-24T07:00:00.000Z</RegistrationDate>
    <Site>US</Site>
    <Status>Confirmed</Status>
    <UserID>Buyer1</UserID>
    <UserIDChanged>false</UserIDChanged>
    <UserIDLastChanged>2004-02-24T07:00:00.000Z</UserIDLastChanged>
    <VATStatus>NoVATTax</VATStatus>
    <BuyerInfo>
      <ShippingAddress>
        <Country>US</Country>
        <PostalCode>95125</PostalCode>
      </ShippingAddress>
    </BuyerInfo>
    <UserAnonymized>false</UserAnonymized>
  </HighBidder>
  <LeadCount>0</LeadCount>
  <MinimumToBid currencyID="USD">1.5</MinimumToBid>
  <QuantitySold>1</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>true</SecondChanceEligible>
  <ListingStatus>Completed</ListingStatus>
</SellingStatus>
```

Example 67-1 Sample ItemMarkedPaid Notification

```
<ShippingDetails>
  <AllowPaymentEdit>true</AllowPaymentEdit>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>PT0S</TimeLeft>
<Title>Test Item{Won/Sold} notify</Title>
<HitCount>0</HitCount>
<GetItFast>false</GetItFast>
<PostalCode>
</PostalCode>
<PictureDetails>
  <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 68

ItemMarkedShipped

eBay sends a **ItemMarkedShipped** notification to the seller whenever a seller who has sold an item from eBay, no matter it is an auction item, a fixed priced store item, a fixed priced item (BIN), or an auction item sold with BestOffer. The typical flow of events is:

A third party subscribes to the **ItemMarkedShipped** notification on behalf of a seller.

A seller marks the item as **Shipped**, using the **CompleteSale** call.

eBay sends the **ItemMarkedShipped** notification to the third party, indicating the seller sold the item before the listing expiration.

Data Fields Returned

The data fields the **ItemMarkedShipped** notification returns are the same as those returned by **GetItemResponse**. Check **GetItemResponse** in the generated WsdlDoc documentation.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

Example 68-1 Sample ItemMarkedShipped Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
    <ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
    xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
        <ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">Z2yhKdKmS0Ga5VPMLOAlg==</
        ebl:NotificationSignature>
    </ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
    <GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
        <Timestamp>2008-02-13T03:47:28.106Z</Timestamp>
        <Ack>Success</Ack>
        <CorrelationID>137541140</CorrelationID>
        <Version>553</Version>
        <Build>e553_core_Bundled_6057805_R1</Build>
        <NotificationEventName>ItemMarkedShipped</NotificationEventName>
        <RecipientUserID>Seller1</RecipientUserID>
        <Item>
            <AutoPay>false</AutoPay>
            <BuyerProtection>ItemEligible</BuyerProtection>
            <BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
            <Country>US</Country>
            <Currency>USD</Currency>
            <GiftIcon>0</GiftIcon>
            <HitCounter>NoHitCounter</HitCounter>
            <ItemID>250000627102</ItemID>
            <ListingDetails>
                <Adult>false</Adult>
                <BindingAuction>false</BindingAuction>
                <CheckoutEnabled>true</CheckoutEnabled>
                <ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
                <ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
                <ConvertedReservePrice currencyID="USD">0.0</ConvertedReservePrice>
                <HasReservePrice>false</HasReservePrice>
                <StartTime>2008-02-13T03:42:45.000Z</StartTime>
                <EndTime>2008-02-13T03:43:32.000Z</EndTime>
                <ViewItemURL>http://cgi.qa-api012.qa.ebay.com/ws/
eBayISAPI.dll?ViewItem&item=250000627102&category=1463</ViewItemURL>
                <HasUnansweredQuestions>false</HasUnansweredQuestions>
                <HasPublicMessages>false</HasPublicMessages>
                <ExpressListing>false</ExpressListing>
                <ViewItemURLForNaturalSearch>http://cgi.qa-api012.qa.ebay.com/Test-Item-Won-Sold-
notify_W0QQitemZ250000627102QQcategoryZ1463QcmdZViewItem</ViewItemURLForNaturalSearch>
            </ListingDetails>
            <ListingDuration>Days_5</ListingDuration>
            <ListingType>Chinese</ListingType>
            <Location>San Jose, CA</Location>
            <PaymentMethods>PaymentSeeDescription</PaymentMethods>
            <PaymentMethods>PayPal</PaymentMethods>
            <PayPalEmailAddress>ve2@aol.com</PayPalEmailAddress>
            <PrimaryCategory>
                <CategoryID>1463</CategoryID>
                <CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
            </PrimaryCategory>
            <PrivateListing>false</PrivateListing>
            <Quantity>1</Quantity>
            <ReservePrice currencyID="USD">0.0</ReservePrice>
            <ReviseStatus>
                <ItemRevised>false</ItemRevised>
            </ReviseStatus>
        </Item>
    </GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Example 68-1 Sample ItemMarkedShipped Notification

```
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>seller@email.com</Email>
  <FeedbackScore>100</FeedbackScore>
  <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>Seller1</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <LiveAuctionAuthorized>false</LiveAuctionAuthorized>
    <MerchandizingPref>OptIn</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerLevel>None</SellerLevel>
    <StoreOwner>false</StoreOwner>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
    <SafePaymentExempt>true</SafePaymentExempt>
  </SellerInfo>
  <MotorsDealer>false</MotorsDealer>
</Seller>
<SellingStatus>
  <BidCount>2</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.25</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.25</CurrentPrice>
  <HighBidder>
    <AboutMePage>false</AboutMePage>
    <EIASToken>nY+sHZ2PrBmdj6wVnY+seZ2PrA2dj6wJkoWoCJaGoAmdj6x9nY+seQ==</EIASToken>
    <Email>buyer@email.com</Email>
    <FeedbackScore>5</FeedbackScore>
    <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
    <FeedbackPrivate>false</FeedbackPrivate>
    <FeedbackRatingStar>None</FeedbackRatingStar>
    <IDVerified>false</IDVerified>
    <eBayGoodStanding>true</eBayGoodStanding>
    <NewUser>false</NewUser>
    <RegistrationDate>2004-02-24T07:00:00.000Z</RegistrationDate>
    <Site>US</Site>
    <Status>Confirmed</Status>
    <UserID>Buyer1</UserID>
    <UserIDChanged>false</UserIDChanged>
    <UserIDLastChanged>2004-02-24T07:00:00.000Z</UserIDLastChanged>
    <VATStatus>NoVATTax</VATStatus>
    <BuyerInfo>
      <ShippingAddress>
        <Country>US</Country>
        <PostalCode>95125</PostalCode>
      </ShippingAddress>
    </BuyerInfo>
    <UserAnonymized>false</UserAnonymized>
  </HighBidder>
  <LeadCount>0</LeadCount>
  <MinimumToBid currencyID="USD">1.5</MinimumToBid>
  <QuantitySold>1</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>true</SecondChanceEligible>
  <ListingStatus>Completed</ListingStatus>
</SellingStatus>
```

Example 68-1 Sample ItemMarkedShipped Notification

```
<ShippingDetails>
  <AllowPaymentEdit>true</AllowPaymentEdit>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>PT0S</TimeLeft>
<Title>Test Item{Won/Sold} notify</Title>
<HitCount>0</HitCount>
<GetItFast>false</GetItFast>
<PostalCode>
</PostalCode>
<PictureDetails>
  <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 69

ItemRemovedFromBidGroup

A **ItemRemovedFromBidGroup** notification is sent to a subscribing user when a buyer deletes a pending item from his or her bid group, as discussed in [Managing Buying and Watch Lists with My eBay](#).

Platform Choice

When a user requests, via the notification preferences web page, to be notified of this event, the default is for the notification to be sent via API ("Platform"). The user can choose instead to be notified via wireless ("SMS").

Data Fields Returned

The data fields the **ItemRemovedFromBidGroup** notification returns are the same as those returned by [GetItemResponse](#) with the default detail level.

See ["SOAP Message Body"](#) on page 628 for details on the standard elements returned for all notifications.

Example

Example 69-1 Sample ItemRemovedFromBidGroup Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">z0jqKt0hyp2EBRnBHBZLhw==</
ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-02-21T02:28:36.093Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>137932190</CorrelationID>
<Version>553</Version>
<Build>e553_core_Bundled_6114677_R1</Build>
<NotificationEventName>ItemRemovedFromBidGroup</NotificationEventName>
<RecipientUserID>Buyer1</RecipientUserID>
<Item>
<AutoPay>false</AutoPay>
<BuyerProtection>ItemEligible</BuyerProtection>
<BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
<Country>US</Country>
<Currency>USD</Currency>
<GiftIcon>0</GiftIcon>
<HitCounter>NoHitCounter</HitCounter>
<ItemID>250000632012</ItemID>
<ListingDetails>
<Adult>false</Adult>
<BindingAuction>false</BindingAuction>
<CheckoutEnabled>true</CheckoutEnabled>
<ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
<ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
<HasReservePrice>false</HasReservePrice>
<StartTime>2008-02-17T01:41:10.000Z</StartTime>
<EndTime>2008-02-22T01:41:10.000Z</EndTime>
<ViewItemURL>http://cgi.qa-api011.qa.ebay.com/ws/
eBayISAPI.dll?ViewItem&item=250000632012&category=1463</ViewItemURL>
<HasUnansweredQuestions>false</HasUnansweredQuestions>
<HasPublicMessages>false</HasPublicMessages>
<BuyItNowAvailable>true</BuyItNowAvailable>
<ExpressListing>false</ExpressListing>
<ViewItemURLForNaturalSearch>http://cgi.qa-api011.qa.ebay.com/Test-Item-AddedTo-RemovedFrom-
BidGroup-1_W0QQitemZ250000632012QQcategoryZ1463QcmdZViewItem</ViewItemURLForNaturalSearch>
</ListingDetails>
<ListingDuration>Days_5</ListingDuration>
<ListingType>Chinese</ListingType>
<Location>San Jose, CA</Location>
<PaymentMethods>PaymentSeeDescription</PaymentMethods>
<PaymentMethods>PayPal</PaymentMethods>
<PrimaryCategory>
<CategoryID>1463</CategoryID>
<CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
</PrimaryCategory>
<PrivateListing>false</PrivateListing>
<Quantity>1</Quantity>
<ReviseStatus>
<ItemRevised>false</ItemRevised>
</ReviseStatus>
```

Example 69-1 Sample ItemRemovedFromBidGroup Notification (Continued)

```
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>Invalid Request</Email>
  <FeedbackScore>100</FeedbackScore>
  <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>Seller1</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <MerchandizingPref>Opt In</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerLevel>None</SellerLevel>
    <StoreOwner>false</StoreOwner>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
    <SafePaymentExempt>true</SafePaymentExempt>
  </SellerInfo>
  <MotorsDealer>false</MotorsDealer>
</Seller>

<SellingStatus>
  <BidCount>0</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.0</CurrentPrice>
  <MinimumToBid currencyID="USD">1.0</MinimumToBid>
  <QuantitySold>0</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>false</SecondChanceEligible>
  <ListingStatus>Active</ListingStatus>
</SellingStatus>

<ShippingDetails>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
</ShippingDetails>

<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>PT23H12M34S</TimeLeft>
<Title>Test Item{AddedTo,RemovedFrom}BidGroup 1</Title>
<HitCount>1</HitCount>
<GetItFast>false</GetItFast>
<PostalCode></PostalCode>
<PictureDetails>
  <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 70

ItemRemovedFromWatchList

eBay sends a **ItemRemovedFromWatchList** notification to third parties subscribed on the buyer's behalf whenever a buyer removes an item from his or her watchlist. The typical flow of events is:

- A third party subscribes to the **ItemRemovedFromWatchList** notification on behalf of a buyer.
- The buyer removes an item from his or her watchlist.
- eBay sends the **ItemRemovedFromWatchList** notification to the third party, indicating that the item has been removed from the watchlist.

eBay also sends **ItemRemovedFromWatchList** notifications to the third party for all items the buyer removes, either through the eBay sites or programmatically from an API or SDK.

Data Fields Returned

The data fields the **ItemRemovedFromWatchList** notification returns are the same as those returned by **GetItemResponse** with the default detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 70-1 Sample ItemRemovedFromWatchList Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">D9D8oIo39sOnjQoCNa5mSw==</
ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-02-13T21:20:13.687Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>137591200</CorrelationID>
<Version>551</Version>
<Build>e551_core_Bundled_6079039_R1</Build>
<NotificationEventName>ItemRemovedFromWatchList</NotificationEventName>
<RecipientUserID>Buyer1</RecipientUserID>
<Item>
<AutoPay>false</AutoPay>
<BuyerProtection>ItemEligible</BuyerProtection>
<BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
<Country>US</Country>
<Currency>USD</Currency>
<GiftIcon>0</GiftIcon>
<HitCounter>NoHitCounter</HitCounter>
<ItemID>250000627649</ItemID>
<ListingDetails>
<Adult>false</Adult>
<BindingAuction>false</BindingAuction>
<CheckoutEnabled>true</CheckoutEnabled>
<ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
<ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
<HasReservePrice>false</HasReservePrice>
<StartTime>2008-02-13T21:18:26.000Z</StartTime>
<EndTime>2008-02-18T21:18:26.000Z</EndTime>
<ViewItemURL>http://cgi.qa-api015.qa.ebay.com/ws/
eBayISAPI.dll?ViewItem&item=250000627649&category=1463</ViewItemURL>
<HasUnansweredQuestions>false</HasUnansweredQuestions>
<HasPublicMessages>false</HasPublicMessages>
<BuyItNowAvailable>true</BuyItNowAvailable>
<ExpressListing>false</ExpressListing>
<ViewItemURLForNaturalSearch>http://cgi.qa-api015.qa.ebay.com/Test-ItemRemovedFromWatchList-
notify_W0QQitemZ250000627649QQcategoryZ1463QQcmdZViewItem</ViewItemURLForNaturalSearch>
</ListingDetails>
```

Example 70-1 Sample ItemRemovedFromWatchList Notification

```
<ListingDuration>Days_5</ListingDuration>
<ListingType>Chinese</ListingType>
<Location>San Jose, CA</Location>
<PaymentMethods>PaymentSeeDescription</PaymentMethods>
<PaymentMethods>PayPal</PaymentMethods>
<PrimaryCategory>
  <CategoryID>1463</CategoryID>
  <CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
</PrimaryCategory>
<PrivateListing>false</PrivateListing>
<Quantity>1</Quantity>
<RevisedStatus>
  <ItemRevised>false</ItemRevised>
</RevisedStatus>
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>Invalid Request</Email>
  <FeedbackScore>100</FeedbackScore>
  <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>Seller1</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <MerchandizingPref>OptIn</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerLevel>None</SellerLevel>
    <StoreOwner>false</StoreOwner>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
    <SafePaymentExempt>true</SafePaymentExempt>
  </SellerInfo>
  <MotorsDealer>false</MotorsDealer>
</Seller>
```

Example 70-1 Sample ItemRemovedFromWatchList Notification

```
<SellingStatus>
  <BidCount>0</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.0</CurrentPrice>
  <MinimumToBid currencyID="USD">1.0</MinimumToBid>
  <QuantitySold>0</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>false</SecondChanceEligible>
  <ListingStatus>Active</ListingStatus>
</SellingStatus>
<ShippingDetails>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>P4DT23H58M13S</TimeLeft>
<Title>Test ItemRemovedFromWatchList notify</Title>
<HitCount>0</HitCount>
<GetItFast>false</GetItFast>
<PostalCode>
</PostalCode>
<PictureDetails>
  <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 71

ItemRevised

When a seller revises an item, eBay sends an **ItemRevised** notification to third parties who are subscribed on the seller's behalf. The typical flow of events is:

- A third party subscribes to the **ItemRevised** notification on behalf of a seller.
- The seller revises an item.
- eBay sends the **ItemRevised** notification to the third party, indicating the item was just revised.
- eBay also sends **ItemRevised** notifications to the third party for all items the seller revises, either through the eBay sites or programmatically from an API or SDK.

Note: If an active item's SKU value/custom label is updated through Selling Manager Pro or through a Selling Manager API call, an ItemRevised notification **will not** be sent to the subscriber. However, there is an upcoming project to remove this limitation in the near future.

The maximum number of **ItemRevised** notifications that eBay can send is one per item (that a third party application is monitoring). If a seller lists an item with the seller's application, and then a second application revises the listing, both applications will receive the notification. This assumes both applications had set their notification preferences for receiving the **ItemRevised** notification using **SetNotificationPreferences**.

Data Fields Returned

The data fields the **ItemRevised** notification returns are the same as those returned by **GetItem** with the default detail level. Check **GetItemResponse** (which contains **Item**) in the generated WSDLDoc documentation.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

Example 71-1 Sample ItemRevised Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0"
xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
4tzLioLLe3eEchFIDFFJba==</ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2005-04-30T03:43:03.029Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>122611410</CorrelationID>
<Version>491</Version>
<Build>e491_core_Bundled_3979444_R1</Build>
<NotificationEventName>ItemRevised</NotificationEventName>
<RecipientUserID>seller1</RecipientUserID>
<Item>
<AutoPay>false</AutoPay>
<BuyerProtection>ItemIneligible</BuyerProtection>
<BuyItNowPrice currencyID="USD">0.0</BuyItNowPrice>
<Country>US</Country>
<Currency>USD</Currency>
<GiftIcon>0</GiftIcon>
<HitCounter>NoHitCounter</HitCounter>
<ItemID>123456789012</ItemID>

<ListingDetails>
<Adult>false</Adult>
<BindingAuction>false</BindingAuction>
<CheckoutEnabled>true</CheckoutEnabled>
<ConvertedBuyItNowPrice currencyID="USD">0.0</ConvertedBuyItNowPrice>
<ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
<ConvertedReservePrice currencyID="USD">0.0</ConvertedReservePrice>
<HasReservePrice>false</HasReservePrice>
<StartTime>2006-12-19T22:06:23.000Z</StartTime>
<EndTime>2006-12-24T22:06:23.000Z</EndTime>
<ViewItemURL>http://cgi.ebay.com/ws/
eBayISAPI.dll?ViewItem&item=123456789012&category=1463</ViewItemURL>
<HasUnansweredQuestions>false</HasUnansweredQuestions>
<HasPublicMessages>false</HasPublicMessages>
</ListingDetails>
<ListingDuration>Days_5</ListingDuration>
<ListingType>Chinese</ListingType>
<Location>San Jose</Location>
<PaymentMethods>PaymentSeeDescription</PaymentMethods>
<PrimaryCategory>
<CategoryID>1463</CategoryID>
<CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
</PrimaryCategory>
<PrivateListing>false</PrivateListing>
<ProximitySearchDetails>
<IncludeInProximitySearch>false</IncludeInProximitySearch>
<PostalCode>
</PostalCode>
</ProximitySearchDetails>
<Quantity>1</Quantity>
<ReservePrice currencyID="USD">0.0</ReservePrice>
<ReviseStatus>
<ItemRevised>true</ItemRevised>
</ReviseStatus>
</Item>
</GetItemResponse>

```

Example 71-1 Sample ItemRevised Notification

```
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>seller1@somewhere.com</Email>
  <FeedbackScore>100</FeedbackScore>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>seller1</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <MerchandizingPref>OptIn</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerLevel>None</SellerLevel>
    <StoreOwner>false</StoreOwner>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
  </SellerInfo>
  <MotorsDealer>false</MotorsDealer>
</Seller>
<SellingStatus>
  <BidCount>0</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.0</CurrentPrice>
  <LeadCount>0</LeadCount>
  <MinimumToBid currencyID="USD">1.0</MinimumToBid>
  <QuantitySold>0</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>false</SecondChanceEligible>
  <ListingStatus>Active</ListingStatus>
</SellingStatus>
<ShippingDetails>
  <AllowPaymentEdit>true</AllowPaymentEdit>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>P4DT23H58M10S</TimeLeft>
<Title>Test ItemRevised notification 2</Title>
<VendorHostedPicture>
  <PhotoDisplay>None</PhotoDisplay>
</VendorHostedPicture>
<HitCount>0</HitCount>
<AnythingPoints>false</AnythingPoints>
<GetItFast>false</GetItFast>
<PostalCode/>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 72

ItemSold

eBay sends a **ItemSold** notification to the seller whenever a seller who has sold an item from eBay, no matter it is an auction item, a fixed priced store item, a fixed priced item (BIN), or an auction item sold with BestOffer. The typical flow of events is:

A third party subscribes to the **ItemSold** notification on behalf of a seller.

A buyer buys the item.

eBay sends the **ItemSold** notification to the third party, indicating the seller sold the item before the listing expiration.

Data Fields Returned

The data fields the **ItemSold** notification returns are the same as those returned by **GetItemResponse**. Check **GetItemResponse** in the generated WsdlDoc documentation.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

Example 72-1 Sample ItemSold Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents" xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
      <ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">Z2yhKdKmS0Ga5VPMLOAlg==</ebl:NotificationSignature>
    </ebl:RequesterCredentials>
  </soapenv:Header>
  <soapenv:Body>
    <GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
      <Timestamp>2008-02-13T03:47:28.106Z</Timestamp>
      <Ack>Success</Ack>
      <CorrelationID>137541140</CorrelationID>
      <Version>553</Version>
      <Build>e553_core_Bundled_6057805_R1</Build>
      <NotificationEventName>ItemSold</NotificationEventName>
      <RecipientUserID>Seller1</RecipientUserID>
      <Item>
        <AutoPay>false</AutoPay>
        <BuyerProtection>ItemEligible</BuyerProtection>
        <BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
        <Country>US</Country>
        <Currency>USD</Currency>
        <GiftIcon>0</GiftIcon>
        <HitCounter>NoHitCounter</HitCounter>
        <ItemID>250000627102</ItemID>
        <ListingDetails>
          <Adult>false</Adult>
          <BindingAuction>false</BindingAuction>
          <CheckoutEnabled>true</CheckoutEnabled>
          <ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
          <ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
          <ConvertedReservePrice currencyID="USD">0.0</ConvertedReservePrice>
          <HasReservePrice>false</HasReservePrice>
          <StartTime>2008-02-13T03:42:45.000Z</StartTime>
          <EndTime>2008-02-13T03:43:32.000Z</EndTime>
          <ViewItemURL>http://cgi.qa-api012.qa.ebay.com/ws/eBayISAPI.dll?ViewItem&item=250000627102&category=1463</ViewItemURL>
          <HasUnansweredQuestions>false</HasUnansweredQuestions>
          <HasPublicMessages>false</HasPublicMessages>
          <ExpressListing>false</ExpressListing>
          <ViewItemURLForNaturalSearch>http://cgi.qa-api012.qa.ebay.com/Test-Item-Won-Sold-notify_WQQitemZ250000627102QQcategoryZ1463QcmdZViewItem</ViewItemURLForNaturalSearch>
        </ListingDetails>
      <ListingDuration>Days_5</ListingDuration>
      <ListingType>Chinese</ListingType>
      <Location>San Jose, CA</Location>
      <PaymentMethods>PaymentSeeDescription</PaymentMethods>
      <PaymentMethods>PayPal</PaymentMethods>
      <PayPalEmailAddress>ve2@aol.com</PayPalEmailAddress>
      <PrimaryCategory>
        <CategoryID>1463</CategoryID>
        <CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
      </PrimaryCategory>
      <PrivateListing>false</PrivateListing>
      <Quantity>1</Quantity>
      <ReservePrice currencyID="USD">0.0</ReservePrice>
      <ReviseStatus>
        <ItemRevised>false</ItemRevised>
      </ReviseStatus>
    </Item>
  </GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Example 72-1 Sample ItemSold Notification

```
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>seller@email.com</Email>
  <FeedbackScore>100</FeedbackScore>
  <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>Seller1</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <MerchandizingPref>Opt In</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerLevel>None</SellerLevel>
    <StoreOwner>false</StoreOwner>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
    <SafePaymentExempt>true</SafePaymentExempt>
  </SellerInfo>
  <MotorsDealer>false</MotorsDealer>
</Seller>
<SellingStatus>
  <BidCount>2</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.25</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.25</CurrentPrice>
  <HighBidder>
    <AboutMePage>false</AboutMePage>
    <EIASToken>nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wJkoWoCJaGoAmdj6x9nY+seQ==</EIASToken>
    <Email>buyer@email.com</Email>
    <FeedbackScore>5</FeedbackScore>
    <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
    <FeedbackPrivate>false</FeedbackPrivate>
    <FeedbackRatingStar>None</FeedbackRatingStar>
    <IDVerified>false</IDVerified>
    <eBayGoodStanding>true</eBayGoodStanding>
    <NewUser>false</NewUser>
    <RegistrationDate>2004-02-24T07:00:00.000Z</RegistrationDate>
    <Site>US</Site>
    <Status>Confirmed</Status>
    <UserID>Buyer1</UserID>
    <UserIDChanged>false</UserIDChanged>
    <UserIDLastChanged>2004-02-24T07:00:00.000Z</UserIDLastChanged>
    <VATStatus>NoVATTax</VATStatus>
    <BuyerInfo>
      <ShippingAddress>
        <Country>US</Country>
        <PostalCode>95125</PostalCode>
      </ShippingAddress>
    </BuyerInfo>
    <UserAnonymized>false</UserAnonymized>
  </HighBidder>
  <LeadCount>0</LeadCount>
  <MinimumToBid currencyID="USD">1.5</MinimumToBid>
  <QuantitySold>1</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>true</SecondChanceEligible>
  <ListingStatus>Completed</ListingStatus>
</SellingStatus>
```

Example 72-1 Sample ItemSold Notification

```
<ShippingDetails>
  <AllowPaymentEdit>true</AllowPaymentEdit>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>PT0S</TimeLeft>
<Title>Test Item{Won/Sold} notify</Title>
<HitCount>0</HitCount>
<GetItFast>false</GetItFast>
<PostalCode>
</PostalCode>
<PictureDetails>
  <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 73

ItemUnsold

eBay sends an **ItemUnsold** notification to the seller when a fixed price listing ends without a buyer purchase. A fixed price listing ends when the listing duration time expires or when the seller ends the listing early through the site or by using the [EndItem](#) or [EndItems](#) calls. .

The typical flow of events is:

- A third party subscribes to the **ItemUnsold** notification on behalf of a seller.
- The fixed price item listing ends without a buyer purchase.
- eBay sends the **ItemUnsold** notification to the third party indicating that the fixed price listing ended without a buyer purchase.

The **ItemUnsold** notification will only be sent to a third party if a multi-quantity, fixed price listing ends with zero buyer purchases, regardless of whether there was still inventory left for the single-variation or multi-variation listing. To check if a fixed price listing had any sales, sellers can call [GetItem](#) and look at the **Item.SellingStatus.QuantitySold** (single variation) or **Variation.SellingStatus.QuantitySold** (multiple variations) fields. If **QuantitySold** is '1' or greater, the The **ItemUnsold** notification will not be sent when the listing ends.

Data Fields Returned

The data fields the **ItemUnsold** notification returns are the same as those returned by [GetItemResponse](#) with the default detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 73-1 Sample ItemUnsold Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents" xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
      <ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">/FxIIdJr9AxztavrRrA+tA==</ebl:NotificationSignature>
    </ebl:RequesterCredentials>
  </soapenv:Header>
  <soapenv:Body>
    <GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
      <Timestamp>2008-02-13T04:09:33.731Z</Timestamp>
      <Ack>Success</Ack>
      <CorrelationID>137541260</CorrelationID>
      <Version>553</Version>
      <Build>e553_core_Bundled_6057805_R1</Build>
      <NotificationEventName>ItemUnsold</NotificationEventName>
      <RecipientUserID>Seller1</RecipientUserID>
      <Item>
        <AutoPay>false</AutoPay>
        <BuyerProtection>ItemEligible</BuyerProtection>
        <BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
        <Country>US</Country>
        <Currency>USD</Currency>
        <GiftIcon>0</GiftIcon>
        <HitCounter>NoHitCounter</HitCounter>
        <ItemID>250000627121</ItemID>
        <ListingDetails>
          <Adult>false</Adult>
          <BindingAuction>false</BindingAuction>
          <CheckoutEnabled>true</CheckoutEnabled>
          <ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
          <ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
          <ConvertedReservePrice currencyID="USD">0.0</ConvertedReservePrice>
          <HasReservePrice>false</HasReservePrice>
          <StartTime>2008-02-16T04:07:52.000Z</StartTime>
          <EndTime>2008-02-13T04:07:52.000Z</EndTime>
          <ViewItemURL>http://cgi.qa-api012.qa.ebay.com/ws/eBayISAPI.dll?ViewItem&item=250000627121&category=1463</ViewItemURL>
          <HasUnansweredQuestions>false</HasUnansweredQuestions>
          <HasPublicMessages>false</HasPublicMessages>
          <BuyItNowAvailable>true</BuyItNowAvailable>
          <ExpressListing>false</ExpressListing>
          <ViewItemURLForNaturalSearch>http://cgi.qa-api012.qa.ebay.com/Test-Norman-ItemUnsold-notify_WQQitemZ250000627121QcategoryZ1463QcmdZViewItem</ViewItemURLForNaturalSearch>
        </ListingDetails>
        <ListingDuration>Days_5</ListingDuration>
        <ListingType>Chinese</ListingType>
        <Location>San Jose, CA</Location>
        <PaymentMethods>PaymentSeeDescription</PaymentMethods>
        <PaymentMethods>PayPal</PaymentMethods>
        <PayPalEmailAddress>example2@email.com</PayPalEmailAddress>
        <PrimaryCategory>
          <CategoryID>1463</CategoryID>
          <CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
        </PrimaryCategory>
        <PrivateListing>false</PrivateListing>
        <Quantity>1</Quantity>
        <ReservePrice currencyID="USD">0.0</ReservePrice>
        <ReviseStatus>
          <ItemRevised>false</ItemRevised>
        </ReviseStatus>
      </Item>
    </GetItemResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 73-1 Sample ItemUnsold Notification

```
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>seller@email.com</Email>
  <FeedbackScore>100</FeedbackScore>
  <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>Seller1</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <LiveAuctionAuthorized>false</LiveAuctionAuthorized>
    <MerchandizingPref>OptIn</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerLevel>None</SellerLevel>
    <StoreOwner>false</StoreOwner>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
    <SafePaymentExempt>true</SafePaymentExempt>
  </SellerInfo>
  <MotorsDealer>false</MotorsDealer>
</Seller>

<SellingStatus>
  <BidCount>0</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.0</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.0</CurrentPrice>
  <LeadCount>0</LeadCount>
  <MinimumToBid currencyID="USD">1.0</MinimumToBid>
  <QuantitySold>0</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>false</SecondChanceEligible>
  <ListingStatus>Completed</ListingStatus>
</SellingStatus>

<ShippingDetails>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
</ShippingDetails>

<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>PT0S</TimeLeft>
<Title>Test ItemUnsold notify</Title>
<HitCount>0</HitCount>
<GetItFast>false</GetItFast>
<PostalCode>
</PostalCode>
<PictureDetails>
  <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```


Chapter 74

ItemWon

eBay sends an **ItemWon** notification to third parties subscribed on the buyer's behalf when a buyer wins an item sold by auction or purchases a fixed price item. The typical flow of events is:

- A third party subscribes to the **ItemWon** notification on behalf of a buyer.
- The buyer wins an item sold by auction by having the highest bid or purchases a fixed price item.
- eBay sends the **ItemWon** notification to the third party, indicating the user just won an auction item or purchased a fixed price item.

eBay also sends **ItemWon** notifications to the third party for all items the buyer wins by auction or fixed price items the buyer purchased, either through the eBay sites or programmatically from an API or SDK.

Data Fields Returned

The data fields the **ItemWon** notification returns are the same as those returned by **GetItemResponse** with the default detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 74-1 Sample ItemWon Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">nt7xW+LMfJloyKy+IpzAdQ==</
ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-02-13T03:47:29.106Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>137541130</CorrelationID>
<Version>553</Version>
<Build>e553_core_Bundled_6057805_R1</Build>
<NotificationEventName>ItemWon</NotificationEventName>
<RecipientUserID>Buyer1</RecipientUserID>
<Item>
<AutoPay>false</AutoPay>
<BuyerProtection>ItemEligible</BuyerProtection>
<BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
<Country>US</Country>
<Currency>USD</Currency>
<GiftIcon>0</GiftIcon>
<HitCounter>NoHitCounter</HitCounter>
<ItemID>250000627102</ItemID>
<ListingDetails>
<Adult>false</Adult>
<BindingAuction>false</BindingAuction>
<CheckoutEnabled>true</CheckoutEnabled>
<ConvertedBuyItNowPrice currencyID="USD">10.0</ConvertedBuyItNowPrice>
<ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
<HasReservePrice>false</HasReservePrice>
<StartTime>2008-02-13T03:42:45.000Z</StartTime>
<EndTime>2008-02-13T03:43:32.000Z</EndTime>
<ViewItemURL>http://cgi.qa-api012.qa.ebay.com/ws/
eBayISAPI.dll?ViewItem&item=250000627102&category=1463</ViewItemURL>
<HasUnansweredQuestions>false</HasUnansweredQuestions>
<HasPublicMessages>false</HasPublicMessages>
<ExpressListing>false</ExpressListing>
<ViewItemURLForNaturalSearch>http://cgi.qa-api012.qa.ebay.com/Test-Item-Won-Sold-
notify_WQQitem250000627102QQcategoryZ1463QcmdZViewItem</ViewItemURLForNaturalSearch>
</ListingDetails>
<ListingDuration>Days_5</ListingDuration>
<ListingType>Chinese</ListingType>
<Location>San Jose, CA</Location>
<PaymentMethods>PaymentSeeDescription</PaymentMethods>
<PaymentMethods>PayPal</PaymentMethods>
<PrimaryCategory>
<CategoryID>1463</CategoryID>
<CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
</PrimaryCategory>
<PrivateListing>false</PrivateListing>
<Quantity>1</Quantity>
<ReviseStatus>
<ItemRevised>false</ItemRevised>
</ReviseStatus>
```

Example 74-1 Sample ItemWon Notification

```
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>seller@email.com</Email>
  <FeedbackScore>100</FeedbackScore>
  <PositiveFeedbackPercent>98.1</PositiveFeedbackPercent>
  <FeedbackPrivate>false</FeedbackPrivate>
  <FeedbackRatingStar>Turquoise</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <eBayGoodStanding>true</eBayGoodStanding>
  <NewUser>false</NewUser>
  <RegistrationDate>2004-02-23T23:50:13.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>Seller1</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2004-02-23T23:48:54.000Z</UserIDLastChanged>
  <VATStatus>NoVATTax</VATStatus>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <MerchandizingPref>Opt In</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerLevel>None</SellerLevel>
    <StoreOwner>false</StoreOwner>
    <ExpressEligible>false</ExpressEligible>
    <ExpressWallet>false</ExpressWallet>
    <SafePaymentExempt>true</SafePaymentExempt>
  </SellerInfo>
  <MotorsDealer>false</MotorsDealer>
</Seller>

<SellingStatus>
  <BidCount>2</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">1.25</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">1.25</CurrentPrice>
  <HighBidder>
    <AboutMePage>false</AboutMePage>
    <EIASToken>nY+shZ2PrBndj6wVnY+sEZ2PrA2dj6wJkoWoCJaGoAmdj6x9nY+seQ==</EIASToken>
    <Email>buyer@email.com</Email>
    <FeedbackScore>5</FeedbackScore>
    <PositiveFeedbackPercent>100.0</PositiveFeedbackPercent>
    <FeedbackPrivate>false</FeedbackPrivate>
    <FeedbackRatingStar>None</FeedbackRatingStar>
    <IDVerified>false</IDVerified>
    <eBayGoodStanding>true</eBayGoodStanding>
    <NewUser>false</NewUser>
    <RegistrationDate>2004-02-24T07:00:00.000Z</RegistrationDate>
    <Site>US</Site>
    <Status>Confirmed</Status>
    <UserID>Buyer1</UserID>
    <UserIDChanged>false</UserIDChanged>
    <UserIDLastChanged>2004-02-24T07:00:00.000Z</UserIDLastChanged>
    <VATStatus>NoVATTax</VATStatus>
    <BuyerInfo>
      <ShippingAddress>
        <Country>US</Country>
        <PostalCode>95125</PostalCode>
      </ShippingAddress>
    </BuyerInfo>
    <UserAnonymized>false</UserAnonymized>
  </HighBidder>
  <MinimumToBid currencyID="USD">1.5</MinimumToBid>
  <QuantitySold>1</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>true</SecondChanceEligible>
  <ListingStatus>Completed</ListingStatus>
</SellingStatus>
```

Example 74-1 Sample ItemWon Notification

```
<ShippingDetails>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ThirdPartyCheckout>false</ThirdPartyCheckout>
  <TaxTable/>
</ShippingDetails>
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>PT0S</TimeLeft>
<Title>Test Item{Won/Sold} notify</Title>
<HitCount>0</HitCount>
<GetItFast>false</GetItFast>
<PostalCode>
</PostalCode>
<PictureDetails>
  <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<ProxyItem>false</ProxyItem>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 75

MyMessages Notification

With [MyMessages](#) notifications, you receive messages when new alerts, eBay messages, or member to member messages (M2M) are received in your My Messages mailbox. These messages can contain only header information, or the full header and message body, depending on the amount of detail you subscribe to. Both buyers and sellers can subscribe to this notification.

Before using this notification, be sure that you have read and understood the information in ["Working with Platform Notifications"](#) on page 623.

The data fields returned in these notifications are the same as those returned by [GetMyMessagesResponse](#). See [GetMyMessagesResponseType](#) in the Type reference.

See ["SOAP Message Body"](#) on page 628 for details on the standard elements returned for all notifications.

- ["Setting MyMessages Notification Preferences"](#) on page 737
- ["MyMessagesAlert"](#) on page 738
- ["MyMessagesAlertHeader"](#) on page 738
- ["MyMessageseBayMessage"](#) on page 739
- ["MyMessageseBayMessageHeader"](#) on page 739
- ["MyMessagesM2MMMessage"](#) on page 740
- ["MyMessagesM2MMMessageHeader"](#) on page 740

Setting MyMessages Notification Preferences

Some special rules apply to subscribing to [MyMessages](#) notifications with [SetNotificationPreferences](#). Though you can choose to be notified of all three types of My Messages mail (alerts, messages from eBay, and member to member (M2M) messages), you can only subscribe to one amount of detail at a time. For example, you cannot subscribe to both [MyMessagesM2MMMessageHeader](#) ([GetMyMessages](#) with a detail level of [ReturnHeaders](#)) and [MyMessagesM2MMMessage](#) ([GetMyMessages](#) with a detail level of [ReturnMessages](#)) at the same time.

Subscribing to a MyMessages Notification

To subscribe to a MyMessages notification at a specified amount of detail (for example, headers instead of the full alert or message body):

- Call [SetNotificationPreferences](#) with an `EventType` value of `MyMessagesM2MMessageHeader` and an `EventEnable` value of `Enable`.

Subscribing to a Different Amount of Detail

To subscribe to a different amount of detail (for example, from `MyMessagesM2MMessageHeader` to `MyMessagesM2MMessage`):

- 1 First call [SetNotificationPreferences](#) with an `EventType` value of `MyMessagesM2MMessageHeader` and an `EventEnable` value of `Disable`.
- 2 Next call [SetNotificationPreferences](#) with an `EventType` value of `MyMessagesM2MMessage` and an `EventEnable` value of `Enable`.

Note: `SetNotificationPreferences` will return an error if you attempt to unsubscribe from one amount of detail and subscribe to another amount of detail in the same call.

MyMessagesAlert

This is a notification type where a specified user or application is notified when an alert is sent to My Messages. This notification type sends a [GetMyMessages](#) response at a detail level of `ReturnMessages`. `MyMessagesAlertHeader` and `MyMessagesAlert` cannot be subscribed to at the same time or specified in the same call.

MyMessagesAlertHeader

This is a notification type where a specified user or application is notified when an alert is sent to My Messages. This notification type sends a [GetMyMessages](#) response at a detail level of `ReturnHeaders`. `MyMessagesAlertHeader` and `MyMessagesAlert` cannot be subscribed to at the same time or specified in the same call.

Example

The following sample shows a `MyMessagesAlertHeader` notification.

Example 75-1 Sample MyMessagesAlertHeader Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebL:NotificationSignature xmlns:ebL="urn:ebay:apis:eBLBaseComponents">ABC123==</ebL:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
```

Example 75-1 Sample MyMessagesAlertHeader Notification (Continued)

```
<soapenv:Body>
<GetMyMessagesResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2005-08-04T21:39:12.514Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>94651830</CorrelationID>
<Version>423</Version>
<Build>e423_core_API_1556612_R1</Build>
<NotificationEventName>MyMessagesAlertHeader</NotificationEventName>
<RecipientUserID>sampleuser</RecipientUserID>

<Alerts>
<Alert>
<Sender>eBay</Sender>
<RecipientUserID>sampleuser</RecipientUserID>
<Subject>Important Alert</Subject>
<Priority>3200</Priority>
<AlertID>100856011</AlertID>
<ResolutionStatus>Unresolved</ResolutionStatus>
<Read>false</Read>
<ReceiveDate>2005-08-04T21:23:09.000Z</ReceiveDate>
<ExpirationDate>2006-04-01T22:00:00.000Z</ExpirationDate>
<IsTimedResolution>false</IsTimedResolution>
<ActionURL>https://resolvethis.com</ActionURL>
<ResponseDetails>
<ResponseEnabled>false</ResponseEnabled>
</ResponseDetails>
<Folder>
<FolderID>0</FolderID>
</Folder>
</Alert>
</Alerts>
</GetMyMessagesResponse>
</soapenv:Body>
</soapenv:Envelope>
```

MyMessageseBayMessage

This is a notification type where a specified user or application is notified when a message from eBay is sent to My Messages. This notification type sends a [GetMyMessages](#) response at a detail level of [ReturnMessages](#). [MyMessageseBayMessageHeader](#) and [MyMessageseBayMessage](#) cannot be subscribed to at the same time or specified in the same call.

MyMessageseBayMessageHeader

This is a notification type where a specified user or application is notified when a message from eBay is sent to My Messages. This notification type sends a [GetMyMessages](#) response at a detail level of [ReturnHeaders](#). [MyMessageseBayMessageHeader](#) and [MyMessageseBayMessage](#) cannot be subscribed to at the same time or specified in the same call.

Example

The following sample shows a [MyMessageseBayMessageHeader](#) notification.

Example 75-2 Sample MyMessageseBayMessageHeader Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0"
xmlns:ns="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">ABC123==</ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetMyMessagesResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2005-08-04T22:52:20.701Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>94654140</CorrelationID>
<Version>423</Version>
<Build>e423_core_API_1556612_R1</Build>
<NotificationEventName>MyMessageseBayMessageHeader</NotificationEventName>
<RecipientUserID>sampleuser</RecipientUserID>
<Messages>
<Message>
<Sender>testuser</Sender>
<RecipientUserID>sampleuser</RecipientUserID>
<Subject>The request with originalMessageId</Subject>
<MessageID>1989403</MessageID>
<Flagged>false</Flagged>
<Read>false</Read>
<ReceiveDate>2005-08-04T22:52:08.000Z</ReceiveDate>
<ExpirationDate>2005-10-03T22:52:08.000Z</ExpirationDate>
<ResponseDetails>
<ResponseEnabled>false</ResponseEnabled>
</ResponseDetails>
<Folder>
<FolderID>0</FolderID>
</Folder>
</Message>
</Messages>
</GetMyMessagesResponse>
</soapenv:Body>
</soapenv:Envelope>
```

MyMessagesM2MMessage

This is a notification type where a specified user or application is notified when a member to member (M2M) message is sent to My Messages. This notification type sends a [GetMyMessages](#) response at a detail level of [ReturnMessages](#). [MyMessagesM2MMessageHeader](#) and [MyMessagesM2MMessage](#) cannot be subscribed to at the same time or specified in the same call.

MyMessagesM2MMessageHeader

This is a notification type where a specified user or application is notified when a member to member (M2M) message is sent to My Messages. This notification type sends a [GetMyMessages](#) response at a

detail level of `ReturnHeaders`. `MyMessagesM2MMassageHeader` and `MyMessagesM2MMassage` cannot be subscribed to at the same time or specified in the same call.

Chapter 76

OutBid

An **Outbid** notification is sent to a subscribing user when another buyer places a higher maximum bid than his own maximum bid and he no longer remains the current high bidder.

Platform Choice

When a user requests, via the notification preferences web page, to be notified of this event, the default is for the notification to be sent via API ("Platform"). The user can choose instead to be notified via wireless ("SMS").

Data Fields Returned

The **Outbid** notification returns the same data as [GetItem Call](#) with the default detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 76-1 Sample OutBid Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0"
xmlns:ns1="urn:ebay:apis:eBLBaseComponents"
xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
k/+qlazH8/qjJlFuBg/g==</ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2005-04-30T03:25:48.108Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>82083110</CorrelationID>
<Version>407</Version>
<Build>20050429131630</Build>
<NotificationEventName>OutBid</NotificationEventName>
<RecipientUserID>buyer1</RecipientUserID>
<Item>
<AutoPay>false</AutoPay>
<BuyerProtection>ItemIneligible</BuyerProtection>
<BuyItNowPrice currencyID="USD">0.0</BuyItNowPrice>
<Country>US</Country>
<Currency>USD</Currency>
<Escrow>None</Escrow>
<GiftIcon>0</GiftIcon>
<HitCounter>NoHitCounter</HitCounter>
<ItemID>704637005</ItemID>
<ListingDetails>
<Adult>false</Adult>
<BindingAuction>false</BindingAuction>
<CheckoutEnabled>true</CheckoutEnabled>
<ConvertedBuyItNowPrice currencyID="USD">0.0</ConvertedBuyItNowPrice>
<ConvertedStartPrice currencyID="USD">1.0</ConvertedStartPrice>
<HasReservePrice>false</HasReservePrice>
<StartTime>2005-04-30T03:25:17.000Z</StartTime>
<EndTime>2005-05-05T03:25:17.000Z</EndTime>
<ViewItemURL>http://cgi.ebay.com/ws/
    eBayISAPI.dll?ViewItem&item=704637005&category=1463</ViewItemURL>
<HasUnansweredQuestions>false</HasUnansweredQuestions>
<HasPublicMessages>false</HasPublicMessages>
</ListingDetails>
<ListingDuration>Days_5</ListingDuration>
<ListingType>Chinese</ListingType>
<Location>San Jose</Location>
<PaymentMethods>PaymentSeeDescription</PaymentMethods>
<PrimaryCategory>
    <CategoryID>1463</CategoryID>
    <CategoryName>Collectibles:Trading Cards:Phone Cards</CategoryName>
</PrimaryCategory>
<PrivateListing>false</PrivateListing>
<ProximitySearchDetails>
    <IncludeInProximitySearch>false</IncludeInProximitySearch>
    <PostalCode>
        </PostalCode>
    </ProximitySearchDetails>
<Quantity>1</Quantity>
<ReviseStatus>
    <ItemRevised>false</ItemRevised>
</ReviseStatus>
```

Example 76-1 Sample OutBid Notification (Continued)

```
<Seller>
  <AboutMePage>false</AboutMePage>
  <Email>Invalid Request</Email>
  <FeedbackScore>2</FeedbackScore>
  <FeedbackRatingStar>None</FeedbackRatingStar>
  <IDVerified>false</IDVerified>
  <NewUser>false</NewUser>
  <RegistrationDate>2004-02-24T00:50:13.000Z</RegistrationDate>
  <Site>US</Site>
  <Status>Confirmed</Status>
  <UserID>seller1</UserID>
  <UserIDChanged>false</UserIDChanged>
  <UserIDLastChanged>2004-02-24T00:48:54.000Z</UserIDLastChanged>
  <SellerInfo>
    <AllowPaymentEdit>true</AllowPaymentEdit>
    <BillingCurrency>USD</BillingCurrency>
    <CheckoutEnabled>true</CheckoutEnabled>
    <CIPBankAccountStored>false</CIPBankAccountStored>
    <GoodStanding>true</GoodStanding>
    <MerchandizingPref>OptIn</MerchandizingPref>
    <QualifiesForB2BVAT>false</QualifiesForB2BVAT>
    <SellerLevel>None</SellerLevel>
    <StoreOwner>false</StoreOwner>
  </SellerInfo>
</Seller>

<SellingStatus>
  <BidCount>2</BidCount>
  <BidIncrement currencyID="USD">0.25</BidIncrement>
  <ConvertedCurrentPrice currencyID="USD">2.25</ConvertedCurrentPrice>
  <CurrentPrice currencyID="USD">2.25</CurrentPrice>
  <HighBidder>
    <AboutMePage>false</AboutMePage>
    <EIASToken>nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wJkoejAJKKoAudj6x9nY+seQ==</EIASToken>
    <Email>Invalid Request</Email>
    <FeedbackScore>0</FeedbackScore>
    <FeedbackRatingStar>None</FeedbackRatingStar>
    <IDVerified>false</IDVerified>
    <NewUser>false</NewUser>
    <RegistrationDate>2004-02-24T00:50:13.000Z</RegistrationDate>
    <Site>US</Site>
    <Status>Confirmed</Status>
    <UserID>buyer2</UserID>
    <UserIDChanged>false</UserIDChanged>
    <UserIDLastChanged>2004-02-24T00:48:54.000Z</UserIDLastChanged>
    <BuyerInfo>
      <ShippingAddress>
        <Country>US</Country>
        <PostalCode>95125</PostalCode>
      </ShippingAddress>
    </BuyerInfo>
  </HighBidder>
  <MinimumToBid currencyID="USD">2.5</MinimumToBid>
  <QuantitySold>0</QuantitySold>
  <ReserveMet>true</ReserveMet>
  <SecondChanceEligible>false</SecondChanceEligible>
</SellingStatus>

<ShippingDetails>
  <AllowPaymentEdit>true</AllowPaymentEdit>
  <ApplyShippingDiscount>false</ApplyShippingDiscount>
  <InsuranceOption>NotOffered</InsuranceOption>
  <SalesTax>
    <SalesTaxPercent>0.0</SalesTaxPercent>
    <ShippingIncludedInTax>false</ShippingIncludedInTax>
  </SalesTax>
  <ShippingServiceOptions/>
</ShippingDetails>
```

Example 76-1 Sample OutBid Notification (Continued)

```
<ShipToLocations>US</ShipToLocations>
<Site>US</Site>
<StartPrice currencyID="USD">1.0</StartPrice>
<TimeLeft>P4DT23H59M30S</TimeLeft>
<Title>Test Auction Title</Title>
<PictureDetails>
    <PhotoDisplay>None</PhotoDisplay>
</PictureDetails>
<AnythingPoints>false</AnythingPoints>
</Item>
</GetItemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 77

eBay Return Notifications

["Return Notification Structure" on page 747](#)

["ReturnCreated" on page 747](#)

["ReturnWaitingForSellerInfo" on page 748](#)

["ReturnSellerInfoOverdue" on page 748](#)

["ReturnShipped" on page 748](#)

["ReturnDelivered" on page 748](#)

["ReturnRefundOverdue" on page 748](#)

["ReturnClosed" on page 748](#)

["ReturnEscalated" on page 749](#)

With eBay Return notifications, buyers and sellers that are part of an eBay return can be notified when the return changes state. The following event notifications can be subscribed to and sent to the buyer and seller involved in an eBay return:

- **ReturnCreated:** notification sent to the subscribed buyer and seller when an eBay return is created.
- **ReturnWaitingForSellerInfo:** notification sent to the subscribed seller when an eBay return is waiting for the seller to provide an Return Merchandise Authorization number and/or an alternative return shipping address to the buyer.
- **ReturnSellerInfoOverdue:** notification sent to the subscribed buyer and seller when seller information (Return Merchandise Authorization number and/or alternative return shipping address) due to the buyer is overdue. The seller is expected to provide requested seller information to the buyer within two business days after the day that the buyer opens the return.
- **ReturnShipped:** notification sent to the subscribed buyer and seller when the item associated with the return is return shipped from the buyer back to the seller.
- **ReturnDelivered:** notification sent to the subscribed buyer and seller when the item associated with the return is received by the seller.
- **ReturnRefundOverdue:** notification sent to the subscribed seller when the seller's refund to the buyer is overdue. The seller is expected to issue a refund to the buyer within five business days after the day that the buyer receives the returned item.
- **ReturnClosed:** notification sent to the subscribed buyer and seller when an eBay return is closed.
- **ReturnEscalated:** notification sent to the subscribed buyer and seller when an eBay return is escalated to an eBay Buyer Protection case. When the buyer successfully escalates an eBay return to an eBay Buyer Protection case, the eBay return is closed.

These notifications are set in the [SetNotificationPreferences](#) call. Before using these notifications, be sure that you have read and understood the information in “[Working with Platform Notifications](#)” on page 623.

Return Notification Structure

The following sample shows the payload for a [ReturnCreated](#) notification.

Example 77-1 Sample ReturnCreated Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://
www.ebay.com/marketplace/resolution/v1/services">
<soapenv:Header/>
<soapenv:Body>
<NotificationEvent>
    <NotificationEventName>ReturnCreated</NotificationEventName>
    <RecipientUserID>cps_seller3</RecipientUserID>
    <ExternalUserData>AABCC .. </ExternalUserData>
    <ReturnId>5000000508</ReturnId>
    <CreationDate>2011-12-05T16:36:10.000Z</CreationDate>
    <OtherPartyId>cps_buyer3</OtherPartyId>
    <OtherPartyRole>BUYER</OtherPartyRole>
    <ReturnStatus>2009-09-07T12:50:06.333Z</ReturnStatus>
    <ReturnStatus>MY_RESPONSE_DUE</ReturnStatus>
    <ReturnGlobalId>EBAY-US</ReturnGlobalId>
</NotificationEvent>
</soapenv:Body>
</soapenv:Envelope>
```

The payload structure for each eBay return notification is essentially the same. The key fields are described below:

- **NotificationEventName:** the name of the notification
- **RecipientUserID:** the eBay user ID for the notification subscriber.
- **ExternalUserData:** this field is pulled from the [SetNotificationPreferences](#) request. See the [SetNotificationPreferences](#) call reference for more information about this field.

For more information on other fields in the payload, see the [getReturnDetail](#) call reference documentation in the Return Management API documentation set.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

ReturnCreated

The [ReturnCreated](#) notification is sent to the subscribed buyer and seller when an eBay return is created. At the time this notification is sent, the buyer or seller will want to look at the [ReturnStatus](#) field to see if their response is due or the other party’s response is due. Buyers and sellers can look at the [status](#) field of the [getUserReturns](#) response to see what type of response is due. Alternatively, sellers can call [getActivityOptions](#) to get the next action to take in the case. For more information on these Return Management API calls, see the Return Management API documentation set.

ReturnWaitingForSellerInfo

The [ReturnWaitingForSellerInfo](#) notification is sent to the subscribed seller when an eBay return is waiting for the seller to provide a Return Merchandise Authorization number and/or an alternative return shipping address to the buyer. Sellers can call [provideSellerInfo](#) to provide a Return Merchandise Authorization number and/or an alternative return shipping address to the buyer. For more information on the [provideSellerInfo](#) API call, see the Return Management API documentation set.

ReturnSellerInfoOverdue

The [ReturnSellerInfoOverdue](#) notification is sent to the subscribed buyer and seller when seller information (Return Merchandise Authorization number and/or alternative return shipping address) due to the buyer is overdue. The seller is expected to provide requested seller information to the buyer within two business days after the day that the buyer opens the return. Sellers can call [provideSellerInfo](#) to provide a Return Merchandise Authorization number and/or an alternative return shipping address to the buyer. For more information on the [provideSellerInfo](#) API call, see the Return Management API documentation set.

ReturnShipped

The [ReturnShipped](#) notification is sent to the subscribed buyer and seller when the item associated with the return is return shipped from the buyer back to the seller.

ReturnDelivered

The [ReturnDelivered](#) notification is sent to the subscribed buyer and seller when the item associated with the return is received by the seller.

ReturnRefundOverdue

The [ReturnRefundOverdue](#) notification is sent to the subscribed seller when the seller's refund to the buyer is overdue. The seller is expected to issue a refund to the buyer within five business days after the day that the buyer receives the returned item. Sellers can call [issueRefund](#) to issue a refund to the buyer. For more information on the [issueRefund](#) API call, see the Return Management API documentation set.

ReturnClosed

The [ReturnClosed](#) notification is sent to the subscribed buyer and seller when an eBay return is closed.

ReturnEscalated

The **ReturnEscalated** notification is sent to the subscribed buyer and seller when an eBay return is escalated to an eBay Buyer Protection case. When the buyer successfully escalates an eBay return to an eBay Buyer Protection case, the eBay return is closed.

Chapter 78

SecondChanceOffer

A **SecondChanceOffer** notification is sent when a listing for an item ends, then, for some reason, the transaction does not close. In these cases, the seller will:

- Close out the past transaction for the item.
- Create the item as a new item.
- Offer the item to the past bidders a second time.

Platform Choice

When a user requests, via the notification preferences web page, to be notified of this event, the default is for the notification to be sent via API ("Platform"). The user can choose instead to be notified via wireless ("SMS").

Data Fields Returned

The **SecondChanceOffer** notification returns the same data as the GetItem call with the default detail level.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Chapter 79

BuyerResponseDispute

The flow of events usually works like this:

- The third party subscribes to the [BuyerResponseDispute](#) notification on behalf of the seller.
- The seller lists an item.
- A buyer wins an auction or commits to buying the item, but has not paid within 7 days after the auction closes.
- The seller files a UPI dispute against the buyer.
- eBay sends a notification to the third party for each response the buyer makes.

Data Fields Returned

The data fields the [BuyerResponseDispute](#) notification returns are the same as those returned by [GetDisputeResponse](#). See [DisputeType](#) in [GetDisputeResponseType](#) in the Type documentation.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

Example 79-1 Sample BuyerResponseDispute Notification

```
<?xml version="1.0" encoding="utf-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" 
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema
    instance">
    <soapenv:Header>
        <RequesterCredentials soapenv:mustUnderstand="0" xmlns="urn:ebay:apis:eBLBaseComponents">
            <NotificationSignature>AtYHDyiYcSwjYUSt+P/78Q==</NotificationSignature>
        </ns:RequesterCredentials>
    </soapenv:Header>
    <soapenv:Body>
        <GetDisputeResponse xmlns="urn:ebay:apis:eBLBaseComponents">
            <Timestamp>
                2004-11-15T23:15:51.609Z
            </Timestamp>
            <Ack>Success</Ack>
            <CorrelationID>
                59234230
            </CorrelationID>
            <Version>387</Version>
            <Build>20041113215337</Build>
            <NotificationEventName>
                UPIBuyerResponseToDispute
            </NotificationEventName>
            <Dispute>
                <DisputeID>197422</DisputeID>
                <DisputeRecordType>UnpaidItem</DisputeRecordType>
                <DisputeState>Closed</DisputeState>
                <DisputeStatus>ClosedFVFCreditNoStrike</DisputeStatus>
                <BuyerUserID>qatuser1</BuyerUserID>
                <SellerUserID>qatuser2</SellerUserID>
                <TransactionID>1894299</TransactionID>
                <Item>
                    <ItemID>4057399802</ItemID>
                    <ListingDetails>
                        <StartTime>2004-11-15T22:57:55.000Z</StartTime>
                        <EndTime>2004-11-15T22:57:57.000Z</EndTime>
                    </ListingDetails>
                    <Quantity>1</Quantity>
                    <SellingStatus>
                        <ConvertedCurrentPrice currencyID="USD">10.0</ConvertedCurrentPrice>
                        <CurrentPrice currencyID="USD">10.0</CurrentPrice>
                    </SellingStatus>
                    <Site>US</Site>
                    <Title>Test add item title</Title>
                </Item>
                <DisputeReason>TransactionMutuallyCanceled</DisputeReason>
                <DisputeExplanation>UnableToResolveTerms</DisputeExplanation>
                <DisputeCreditEligibility>InEligible</DisputeCreditEligibility>
                <DisputeCreatedTime>2004-11-15T22:58:00.000Z</DisputeCreatedTime>
                <DisputeModifiedTime>2004-11-15T22:58:00.000Z</DisputeModifiedTime>
                <DisputeResolution>
                    <DisputeResolutionRecordType>FVFCredit</DisputeResolutionRecordType>
                    <DisputeResolutionReason>Unresolved</DisputeResolutionReason>
                    <ResolutionTime>2004-11-15T22:58:00.000Z</ResolutionTime>
                </DisputeResolution>
                <DisputeMessage>
                    <MessageID>394659</MessageID>
                    <MessageSource>eBay</MessageSource>
                    <MessageCreationTime>2004-11-15T22:58:00.000Z</MessageCreationTime>
                    <MessageText>An Unpaid Item dispute has been opened for the following item:
                        Test add item title (#4057399802)
                        Reason given for Unpaid Item: We have both agreed not to complete the transaction.
                        Buyer actions reported by seller: We were unable to resolve a disagreement
                        over terms.</MessageText>
                </DisputeMessage>
            </Dispute>
        </GetDisputeResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Example 79-1 Sample BuyerResponseDispute Notification (Continued)

```
<DisputeMessage>
  <MessageID>394660</MessageID>
  <MessageSource>Buyer</MessageSource>
  <MessageCreationTime>2004-11-15T22:58:00.000Z</MessageCreationTime>
  <MessageText>UPI BuyerResponseToDispute notification testing</MessageText>
</DisputeMessage>
<DisputeMessage>
  <MessageID>394661</MessageID>
  <MessageSource>eBay</MessageSource>
  <MessageCreationTime>2004-11-15T22:58:00.000Z</MessageCreationTime>
  <MessageText>
    The Unpaid Item dispute has been closed for the following reason:
    We've mutually agreed not to complete the transaction.
  </MessageText>
</DisputeMessage>
</Dispute>
</GetDisputeResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 80

SellerClosedDispute

eBay sends a **SellerClosedDispute** notification to a subscribing user when the seller closes a dispute.

Data Fields Returned

The **SellerClosedDispute** notification looks like a response from [GetDispute](#). In the response, the **NotificationEventName** element has the value **SellerClosedDispute**.

The main body of the notification is a **Dispute** element. See the Type documentation for the complex type **DisputeType**.

Also see “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

The following example shows a **SellerClosedDispute** notification.

Example 80-1 Sample SellerClosedDispute Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
  <ebay:NotificationSignature xmlns:ebay="https://developer.ebay.com/2001/schema">
    ae5f88f3f872555a505fed9ede168cf0
  </ebay:NotificationSignature>
</soapenv:Header>
<soapenv:Body>
  <GetDisputeResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2005-02-02T02:45:26.092Z</Timestamp>
    <Ack>Success</Ack>
    <CorrelationID>
      00000000-00000000-00000000-00000000-00000000-0000000000
    </CorrelationID>
    <Version>393</Version>
    <Build>20050126013939</Build>
    <NotificationEventName>SellerClosedDispute</NotificationEventName>
    <Dispute>
      <DisputeID>198928</DisputeID>
      <DisputeRecordType>UnpaidItem</DisputeRecordType>
      <DisputeState>Closed</DisputeState>
      <DisputeStatus>ClosedNoFVFCreditNoStrike</DisputeStatus>
      <BuyerUserID>abc123</BuyerUserID>
      <SellerUserID>xyz789</SellerUserID>
      <TransactionID>863318</TransactionID>
    </Dispute>
  </GetDisputeResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Example 80-1 Sample SellerClosedDispute Notification (Continued)

```
<Item>
  <ItemID>2206653838</ItemID>
  <ListingDetails>
    <StartTime>2004-11-23T17:35:19.000Z</StartTime>
    <EndTime>2004-11-23T17:35:23.000Z</EndTime>
  </ListingDetails>
  <Quantity>1</Quantity>
  <SellingStatus>
    <ConvertedCurrentPrice currencyID="USD">34.0</ConvertedCurrentPrice>
    <CurrentPrice currencyID="USD">34.0</CurrentPrice>
  </SellingStatus>
  <Site>US</Site>
  <Title>Test add item</Title>
</Item>

<DisputeReason>BuyerHasNotPaid</DisputeReason>
<DisputeExplanation>BuyerHasNotResponded</DisputeExplanation>
<DisputeCreditEligibility>InEligible</DisputeCreditEligibility>
<DisputeCreatedTime>2004-12-01T17:35:42.000Z</DisputeCreatedTime>
<DisputeModifiedTime>2005-01-24T21:25:23.000Z</DisputeModifiedTime>

<DisputeMessage>
  <MessageID>398788</MessageID>
  <MessageSource>eBay</MessageSource>
  <MessageCreationTime>2004-12-01T17:35:42.000Z</MessageCreationTime>
  <MessageText>
    An Unpaid Item dispute has been opened for the following item:
    Test add item (#2206653838)
    Reason given for Unpaid Item: The buyer has not paid for the item.
    Buyer actions reported by seller: The buyer has not responded.
  </MessageText>
</DisputeMessage>

<DisputeMessage>
  <MessageID>412279</MessageID>
  <MessageSource>eBay</MessageSource>
  <MessageCreationTime>2005-01-24T21:21:24.000Z</MessageCreationTime>
  <MessageText>
    The Unpaid Item dispute has been closed for the following reason:
    The reporting period has elapsed.</MessageText>
</DisputeMessage>

</Dispute>
</GetDisputeResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 81

SellerOpenedDispute

eBay sends a **SellerOpenedDispute** notification to a subscribing user when the seller opens a dispute.

Data Fields Returned

The **SellerOpenedDispute** notification looks like a response from **GetDispute**. In the response, the **NotificationEventName** element has the value **SellerOpenedDispute**.

The main body of the notification is a **Dispute** element with the complex type **DisputeType**. You can view its entire structure in the Type reference:

[Type Reference](#)

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

The following example shows a **SellerOpenedDispute** notification.

Example 81-1 Sample SellerOpenedDispute Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
  < ebay:NotificationSignature xmlns:ebay="https://developer.ebay.com/2001/schema">
    ae5f88f3f872555a505fed9ede168cf0
  </ ebay:NotificationSignature>
</soapenv:Header>
<soapenv:Body>
  <GetDisputeResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2005-02-02T02:45:26.092Z</Timestamp>
    <Ack>Success</Ack>
    <CorrelationID>
      00000000-00000000-00000000-00000000-00000000-0000000000
    </CorrelationID>
    <Version>393</Version>
    <Build>20050126013939</Build>
    <NotificationEventName>SellerOpenedDispute</NotificationEventName>
    <Dispute>
      <DisputeID>198928</DisputeID>
      <DisputeRecordType>UnpaidItem</DisputeRecordType>
      <DisputeState>Closed</DisputeState>
      <DisputeStatus>ClosedNoFVFCreditNoStrike</DisputeStatus>
      <BuyerUserID>abc123</BuyerUserID>
      <SellerUserID>xyz789</SellerUserID>
      <TransactionID>863318</TransactionID>
    </Dispute>
  </GetDisputeResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Example 81-1 Sample SellerOpenedDispute Notification (Continued)

```
<Item>
  <ItemID>2206653838</ItemID>
  <ListingDetails>
    <StartTime>2004-11-23T17:35:19.000Z</StartTime>
    <EndTime>2004-11-23T17:35:23.000Z</EndTime>
  </ListingDetails>
  <Quantity>1</Quantity>
  <SellingStatus>
    <ConvertedCurrentPrice currencyID="USD">34.0</ConvertedCurrentPrice>
    <CurrentPrice currencyID="USD">34.0</CurrentPrice>
  </SellingStatus>
  <Site>US</Site>
  <Title>My great item thingy</Title>
</Item>

<DisputeReason>BuyerHasNotPaid</DisputeReason>
<DisputeExplanation>BuyerHasNotResponded</DisputeExplanation>
<DisputeCreditEligibility>InEligible</DisputeCreditEligibility>
<DisputeCreatedTime>2004-12-01T17:35:42.000Z</DisputeCreatedTime>
<DisputeModifiedTime>2005-01-24T21:25:23.000Z</DisputeModifiedTime>

<DisputeMessage>
  <MessageID>398788</MessageID>
  <MessageSource>eBay</MessageSource>
  <MessageCreationTime>2004-12-01T17:35:42.000Z</MessageCreationTime>
  <MessageText>
    An Unpaid Item dispute has been opened for the following item:
    Test add item (#2206653838)
    Reason given for Unpaid Item: The buyer has not paid for the item.
    Buyer actions reported by seller: The buyer has not responded.
  </MessageText>
</DisputeMessage>

<DisputeMessage>
  <MessageID>412279</MessageID>
  <MessageSource>eBay</MessageSource>
  <MessageCreationTime>2005-01-24T21:21:24.000Z</MessageCreationTime>
  <MessageText>
    The Unpaid Item dispute has been closed for the following reason:
    The reporting period has elapsed.</MessageText>
</DisputeMessage>

</Dispute>
</GetDisputeResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 82

Seller Payout Notifications

- “ReadyToShip” on page 758
- “ReadyForPayout” on page 759
- “UnmatchedPaymentReceived” on page 759
- “RefundSuccess” on page 759
- “RefundFailure” on page 759

German and Austrian sellers can subscribe to the following event notifications:

- **ReadyToShip**: this notification is sent to the subscribed DE or AT seller when an order (subject to the new eBay payment process) has been paid for by the buyer.
- **ReadyForPayout**: this notification is sent to the subscribed DE or AT seller when eBay is scheduled to distribute the payout to the seller's account in the next disbursement cycle.
- **UnmatchedPaymentReceived**: this notification is sent to the subscribed DE or AT seller when the buyer's EFT payment for an order does not match the original amount at order creation time.
- **RefundSuccess**: this notification is sent to the subscribed DE or AT seller when the seller's voluntary refund (using the site or the **issueRefund** call of the Order Adjustment API) to the buyer is successful.
- **RefundFailure**: this notification is sent to the subscribed DE or AT seller when the seller's voluntary refund (using the site or the **issueRefund** call of the Order Adjustment API) to the buyer fails.

These notifications are set in the **SetNotificationPreferences** call. Before using these notifications, be sure that you have read and understood the information in “[Working with Platform Notifications](#)” on page 623.

ReadyToShip

The **ReadyToShip** notification is sent to the subscriber (seller) when an order (subject to the new eBay payment process) has been paid for by the buyer. This notification can only be sent to the seller. This notification is triggered and sent when **OrderStatus** is **Completed** in **GetOrders** and other order retrieval API calls. Shipping the item and providing proof of shipment can expedite the process of eBay releasing funds to the seller's account. Starting in late August 2011, sellers new to listing on the German (DE) and Austrian (AT) sites became subject to the new eBay payment process, and beginning in May 2012, all DE and AT sellers are subject to the new eBay payment process.

The **GetOrders** API response is in the notification payload.

ReadyForPayout

The **ReadyForPayout** notification is sent to a DE or AT seller when eBay is scheduled to distribute the payout to the seller's account in the next disbursement cycle. This notification is triggered and sent when the value of **PaymentHoldStatus** is **Released** in **GetOrders** and other order retrieval API calls. Starting in late August 2011, sellers new to listing on the German (DE) and Austrian (AT) sites became subject to the new eBay payment process, and beginning in May 2012, all DE and AT sellers are subject to the new eBay payment process.

The **GetOrders** API response is in the notification payload.

UnmatchedPaymentReceived

The **UnmatchedPaymentReceived** notification is sent to the subscribed DE or AT seller when the buyer's EFT payment for an order does not match the original amount at order creation time. If the seller receives an unmatched payment, the seller must accept or reject the over- or under-payment using the site or using the **acceptAdjustment** or **rejectAdjustment** calls of the Order Adjustment API. See the Order Adjustment API documentation set for more information.

RefundSuccess

The **RefundSuccess** notification is sent to the subscribed DE or AT seller when the seller's voluntary refund (using the site or the **issueRefund** call of the Order Adjustment API) to the buyer is successful. See the Order Adjustment API documentation set for more information.

RefundFailure

The **RefundFailure** notification is sent to the subscribed DE or AT seller when the seller's voluntary refund (using the site or the **issueRefund** call of the Order Adjustment API) to the buyer is successful. See the Order Adjustment API documentation set for more information.

Chapter 83

SellerRespondedToDispute

eBay sends a **SellerRespondedToDispute** notification to a subscribing user when the seller responds to a dispute.

Data Fields Returned

The **SellerRespondedToDispute** notification looks like a response from **GetDispute**. In the response, the **NotificationEventName** element has the value **SellerRespondedToDispute**.

The main body of the notification is a **Dispute** element with the complex type **DisputeType**.

See “[SOAP Message Body](#)” on page 628 for details on the standard elements returned for all notifications.

Example

The following example shows a **SellerRespondedToDispute** notification.

Example 83-1 Sample SellerRespondedToDispute Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
  <ebay:NotificationSignature xmlns:ebay="https://developer.ebay.com/2001/schema">
    ae5f88f3f872555a505fed9ede168cf0
  </ebay:NotificationSignature>
</soapenv:Header>
<soapenv:Body>
  <GetDisputeResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2005-02-02T02:45:26.092Z</Timestamp>
    <Ack>Success</Ack>
    <CorrelationID>
      00000000-00000000-00000000-00000000-00000000-0000000000
    </CorrelationID>
    <Version>393</Version>
    <Build>20050126013939</Build>
    <NotificationEventName>SellerRespondedToDispute</NotificationEventName>
    <Dispute>
      <DisputeID>198928</DisputeID>
      <DisputeRecordType>UnpaidItem</DisputeRecordType>
      <DisputeState>Closed</DisputeState>
      <DisputeStatus>ClosedNoFVFCreditNoStrike</DisputeStatus>
      <BuyerUserID>abc123</BuyerUserID>
      <SellerUserID>xyz789</SellerUserID>
      <TransactionID>863318</TransactionID>
```

Example 83-1 Sample SellerRespondedToDispute Notification (Continued)

```
<Item>
  <ItemID>2206653838</ItemID>
  <ListingDetails>
    <StartTime>2004-11-23T17:35:19.000Z</StartTime>
    <EndTime>2004-11-23T17:35:23.000Z</EndTime>
  </ListingDetails>
  <Quantity>1</Quantity>
  <SellingStatus>
    <ConvertedCurrentPrice currencyID="USD">34.0</ConvertedCurrentPrice>
    <CurrentPrice currencyID="USD">34.0</CurrentPrice>
  </SellingStatus>
  <Site>US</Site>
  <Title>Test add item</Title>
</Item>

<DisputeReason>BuyerHasNotPaid</DisputeReason>
<DisputeExplanation>BuyerHasNotResponded</DisputeExplanation>
<DisputeCreditEligibility>InEligible</DisputeCreditEligibility>
<DisputeCreatedTime>2004-12-01T17:35:42.000Z</DisputeCreatedTime>
<DisputeModifiedTime>2005-01-24T21:25:23.000Z</DisputeModifiedTime>

<DisputeMessage>
  <MessageID>398788</MessageID>
  <MessageSource>eBay</MessageSource>
  <MessageCreationTime>2004-12-01T17:35:42.000Z</MessageCreationTime>
  <MessageText>
    An Unpaid Item dispute has been opened for the following item:
    Test add item (#2206653838)
    Reason given for Unpaid Item: The buyer has not paid for the item.
    Buyer actions reported by seller: The buyer has not responded.
  </MessageText>
</DisputeMessage>

<DisputeMessage>
  <MessageID>412279</MessageID>
  <MessageSource>eBay</MessageSource>
  <MessageCreationTime>2005-01-24T21:21:24.000Z</MessageCreationTime>
  <MessageText>
    The Unpaid Item dispute has been closed for the following reason:
    The reporting period has elapsed.</MessageText>
</DisputeMessage>

</Dispute>
</GetDisputeResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 84

TokenRevocation

eBay sends a **TokenRevocation** notification to the subscriber when an application revokes a user token. The typical flow of events is:

A third-party application subscribes to the **TokenRevocation** notification on behalf of a user.

The application or the user revokes the token, before it is due to expire, using the **RevokeToken** call.

eBay sends the **TokenRevocation** notification to the subscriber.

Data Fields Returned

The data fields the **TokenRevocation** notification returns are the same as those returned by **GetTokenStatus**. Check **GetTokenStatus** in the API Call Reference Guide.

See “SOAP Message Body” on page 628 for details on the standard elements returned for all notifications.

Example

Example 84-1 Sample TokenRevocation Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ebl:RequesterCredentials soapenv:mustUnderstand="0" xmlns:ns="urn:ebay:apis:eBLBaseComponents" xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
<ebl:NotificationSignature xmlns:ebl="urn:ebay:apis:eBLBaseComponents">X4lbGocDIPUeWiyZ0ONFog==</ebl:NotificationSignature>
</ebl:RequesterCredentials>
</soapenv:Header>
<soapenv:Body>
<GetTokenStatusResponse xmlns="urn:ebay:apis:eBLBaseComponents">
<Timestamp>2008-03-18T23:37:44.055Z</Timestamp>
<Ack>Success</Ack>
<CorrelationID>138758970</CorrelationID>
<Version>557</Version>
<Build>e557_core_Bundled_6282437_R1</Build>
<NotificationEventName>TokenRevocation</NotificationEventName>
<RecipientUserID>bestuser1</RecipientUserID>
<TokenStatus>
<Status>RevokedByeBay</Status>
<EIAToken>EIAToken</EIAToken>
<ExpirationTime>2008-04-04T13:30:38.000Z</ExpirationTime>
<RevocationTime>2008-03-18T21:30:40.000Z</RevocationTime>
</TokenStatus>
</GetTokenStatusResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Chapter 85

WatchedItemEndingSoon

A **WatchedItemEndingSoon** notification is sent to a subscribing user a few minutes before the listing ends, where the number of minutes is set by the user.

Platform Choice

When a user requests, via the notification preferences web page, to be notified of this event, the default is for the notification to be sent via API ("Platform"). The user can choose instead to be notified via wireless ("SMS").

See "[Additional Steps for WatchedItemEndingSoon Subscription](#)" on page 635.

Data Fields Returned

The data fields the **WatchedItemEndingSoon** notification returns are essentially the same as those returned by **GetItem**. The **ItemArray** container will contain at most one **Item** container. If there is no item matching the submitted **ItemID**, no **ItemArray** container will be returned. If the listing ended, **TimeLeft** will be **PT0S**.

See "[SOAP Message Body](#)" on page 628 for details on the standard elements returned for all notifications.

Example

Example 85-1 Sample WatchedItemEndingSoon Notification

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Header>
        <ebl:RequesterCredentials soapenv:mustUnderstand="0">
            xmlns:ns="urn:ebay:apis:eBLBaseComponents"
            xmlns:ebl="urn:ebay:apis:eBLBaseComponents">
                <ebl:NotificationSignature>abc123==</ebl:NotificationSignature>
            </ebl:RequesterCredentials>
        </soapenv:Header>
        <soapenv:Body>
            <GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
                <Timestamp>2007-03-09T19:18:08.284Z</Timestamp>
                <Ack>Success</Ack>
                <Version>505</Version>
                <Build>e505_core_API_32243_90</Build>
                <NotificationEventName>WatchedItemEndingSoon</NotificationEventName>
                <WatchList>
                    <ItemArray>
                        <Item>
                            <BuyItNowPrice currencyID="USD">10.0</BuyItNowPrice>
                            <ItemID>4100613511</ItemID>
                            <ListingDetails>
                                <StartTime>2007-03-08T20:57:14.000Z</StartTime>
                                <EndTime>2007-03-15T20:57:14.000Z</EndTime>
                            </ListingDetails>
                            <Seller>
                                <UserID>sampleuser</UserID>
                            </Seller>
                            <SellingStatus>
                                <CurrentPrice currencyID="USD">10.0</CurrentPrice>
                            </SellingStatus>
                            <Site>US</Site>
                            <TimeLeft>P6DT1H39M6S</TimeLeft>
                            <Title>SDK item title</Title>
                            <QuantityAvailable>1</QuantityAvailable>
                        </Item>
                    </ItemArray>
                    <PaginationResult>
                        <TotalNumberOfPages>1</TotalNumberOfPages>
                        <TotalNumberOfEntries>1</TotalNumberOfEntries>
                    </PaginationResult>
                </WatchList>
            </GetItemResponse>
        </soapenv:Body>
    </soapenv:Envelope>
```

Part XI

API Call Reference Guide

Overview of Calls	770
AddDispute	771
AddDisputeResponse	775
AddFixedPriceItem	781
AddItem	782
AddItems	783
AddMemberMessageAAQToPartner	784
AddMemberMessageRTQ	785
AddMemberMessagesAAQToBidder	786
AddOrder	787
AddSecondChanceItem	788
AddToItemDescription	789
AddToWatchList	790
AddTransactionConfirmationItem	791
CompleteSale	792
ConfirmIdentity	793

DeleteMyMessages	794
EndFixedPriceItem	795
EndItem	796
EndItems	797
FetchToken	798
GetAccount	799
GetAdFormatLeads	800
GetAllBidders	801
GetApiAccessRules	802
GetBestOffers	803
GetBidderList	804
GetCategories	805
GetCategory2CS	806
GetCategory2FinanceOffer	807
GetCategoryFeatures	808
GetCategoryListings	809
GetCategoryMappings	810
GetCategorySpecifics	811
GetChallengeToken	812
GetCharities	813
GetClientAlertsAuthToken	824
GetContextualKeywords	825
GetCrossPromotions	826
GetDescriptionTemplates	827
GetDispute	828
GeteBayDetails	829
GeteBayOfficialTime	830
GetFeedback	831
GetFinanceOffers	832
GetItem	833
GetItemsAwaitingFeedback	834

GetItemShipping	835
GetItemTransactions	836
GetMemberMessages	837
GetMessagePreferences	838
GetMyeBayBuying	839
GetMyeBayReminders	840
GetMyeBaySelling	841
GetMyMessages	842
GetNotificationPreferences	843
GetNotificationsUsage	849
GetOrders	852
GetOrderTransactions	853
GetPromotionalSaleDetails	854
GetPromotionRules	855
GetSellerDashboard	857
GetSellerEvents	858
GetSellerList	859
GetSellerPayments	860
GetSellerTransactions	865
GetSessionID	866
GetShippingDiscountProfiles	867
GetStore	868
GetStoreCategoryUpdateStatus	869
GetStoreCustomPage	870
GetStoreOptions	871
GetStorePreferences	872
GetSuggestedCategories	873
GetTaxTable	874
GetTokenStatus	875
GetUser	876
GetUserContactDetails	877
GetUserDisputes	878

GetUserPreferences	879
GetVeROReasonCodeDetails	880
GetVeROReportStatus	884
GetWantItNowPost	889
GetWantItNowSearchResults	893
IssueRefund	900
LeaveFeedback	903
PlaceOffer	904
RelistFixedPriceItem	905
RelistItem	906
RemoveFromWatchList	907
RespondToBestOffer	908
RespondToFeedback	912
RespondToWantItNowPost	913
ReviseCheckoutStatus	916
ReviseFixedPriceItem	917
ReviseItem	918
ReviseMyMessages	919
ReviseMyMessagesFolders	920
RevokeToken	921
SellerReverseDispute	922
SendInvoice	927
SetMessagePreferences	928
SetNotificationPreferences	929
SetPromotionalSale	930
SetPromotionalSaleListings	931
SetShippingDiscountProfiles	933
SetStore	934
SetStoreCategories	935
SetStoreCustomPage	936
SetStorePreferences	937

SetTaxTable	938
SetUserNotes	939
SetUserPreferences	940
UploadSiteHostedPictures	941
ValidateChallengeInput	942
ValidateTestUserRegistration	943
VerifyAddFixedPriceItem	944
VerifyAddItem	945
VerifyAddSecondChanceItem	946
VeROReportItems	948
Schema Reference Documentation	967

Chapter 86

Overview of Calls

Formerly, this topic contained descriptions of API calls and links to reference information.

These descriptions and links can be found in the [eBay Schema Input/Output Reference](#):

[eBay Schema Input/Output Reference](#)

The [eBay Schema Input/Output Reference](#) contains information about input and output fields for each call, including definitions and validation rules.

For additional information about reference documentation, see “[Schema Reference Documentation](#)” on page 967. For information about planning ahead, suggested best practices, and important information about eBay’s release cycle, please see “[Developing an eBay-Enabled Application](#)” on page 56.

Chapter 87

AddDispute

AddDispute is part of the Unpaid Item Dispute process. See “[Unpaid Item Disputes](#)” on page 455. (It is not used for Item Not Received disputes. See “[Buyer Disputes](#)” on page 476.)

- “[AddDispute Input Fields](#)” on page 771
- “[AddDispute Output Fields](#)” on page 771
- “[AddDispute Details](#)” on page 772
- “[AddDispute Best Practices](#)” on page 772
- “[Testing AddDispute](#)” on page 772
- “[AddDispute Sample](#)” on page 773
- “[Related Calls](#)” on page 773
- “[Change History](#)” on page 774

AddDispute Input Fields

See the following document for information about the call’s fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

AddDispute Output Fields

If the call to **AddDispute** is successful, it returns a dispute ID for the new dispute. If unsuccessful, it returns an error construct (see “[Error Handling](#)” on page 107). If a warning occurs, the result set contains both normal data and an error construct.

See the schema documentation for **AddDispute** for a complete list of its properties, their definitions , and validation rules.

See the following document for information about the call’s fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

AddDispute Details

Working with the Response

If [AddDispute](#) executes successfully, it returns a new dispute ID. You need the dispute ID for other calls that handle disputes, such as [AddDisputeResponse](#). You can also retrieve dispute IDs via [GetUserDisputes](#), as described in “[Getting Details About a Dispute](#)” on page 469.

AddDispute Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

[AddDispute](#) returns only a dispute ID and a call status.

Detail Levels

[AddDispute](#) does not support multiple detail levels. It is not necessary to specify a detail level for this call.

Pagination

[AddDispute](#) does not require pagination.

Data Filters

[AddDispute](#) does not provide data filters for reducing the volume of the response.

Reducing the Number of Calls You Make

Each call to [AddDispute](#) takes one item ID and one transaction ID and creates one dispute. It is not necessary to reduce the number of calls to [AddDispute](#).

Testing AddDispute

- 1 Set up at least one buyer and one seller in the Sandbox environment.
- 2 Have the seller list an item. This gives you an item ID.
- 3 Create a transaction by using the simulated eBay site in the Sandbox. Have the buyer commit to purchase the item, but do not have the buyer pay.
- 4 Get the transaction ID by calling [GetItemTransactions](#) with the item ID.
- 5 Wait at least 7 days for the transaction to age, with no buyer response. (Alternately, you can open a Support Ticket and, for a fee, eBay will artificially age your listing for you.)
- 6 On all calls to [AddDispute](#), be sure to authenticate as the seller.
- 7 Call [AddDispute](#) for the seller, using the item ID and transaction ID. Store the dispute ID that is returned.

- 8 Verify that the dispute ID is valid by adding a comment to the dispute with [AddDisputeResponse](#). Use a dispute activity ID of [SellerAddInformation](#) and a non-empty message text.

AddDispute Sample

The sample in this section was created using the XML API. If you are working with the SOAP API, you can still use this sample as a guide, because it is equivalent to the structure in the body of a SOAP request and response.

Please note that this sample was created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if this sample does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in the sample.

[Example 87-1](#) shows how to open an Unpaid Item dispute with [AddDispute](#). The caller must be the seller and have a valid [ItemID](#) and [TransactionID](#).

Example 87-1 AddDispute Sample Input

```
<?xml version="1.0" encoding="utf-8"?>
<AddDisputeRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <SiteID>0</SiteID>
  <ItemID>9202501086</ItemID>
  <TransactionID>748970</TransactionID>
  <DisputeReason>BuyerHasNotPaid</DisputeReason>
  <DisputeExplanation>BuyerHasNotResponded</DisputeExplanation>
</AddDisputeRequest>
```

[Example 87-2](#) shows the DisputeID in the response.

Example 87-2 AddDispute Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<AddDisputeResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-03-10T02:50:41.004Z</Timestamp>
  <Ack>Success</Ack>
  <CorrelationID>
    00000000-00000000-00000000-00000000-00000000-00000000-00000000
  </CorrelationID>
  <Version>401</Version>
  <Build>20050308050919</Build>
  <DisputeID>213833</DisputeID>
</AddDisputeResponse>
```

Related Calls

- “[AddDisputeResponse](#)” on page 775
- “[GetDispute](#)” on page 828
- “ [GetUserDisputes](#)” on page 878
- “ [SellerReverseDispute](#)” on page 922

Change History

For information about changes that have been made to this call, see the call's "Change History" section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 88

AddDisputeResponse

[AddDisputeResponse](#) is useful for both the Unpaid Item Dispute process (see “[Unpaid Item Disputes](#)” on page 455) and the Item Not Received process (see “[INR \(ItemNotReceived\) Notifications](#)” on page 685) for adding a comment to or closing a dispute.

- “[Overview](#)” on page 775
- “[AddDisputeResponse Input Fields](#)” on page 776
- “[AddDisputeResponse Output Fields](#)” on page 777
- “[AddDisputeResponse Details](#)” on page 777
- “[AddDisputeResponse Best Practices](#)” on page 777
- “[Testing AddDisputeResponse](#)” on page 778
- “[AddDisputeResponse Samples](#)” on page 778
- “[Related Calls](#)” on page 780
- “[Change History](#)” on page 780

Overview

Unpaid Item Disputes

See “[Communicating with the Buyer](#)” on page 466 and “[Closing an Unpaid Item Dispute](#)” on page 467.

Item Not Received Disputes

See also “[Buyer Disputes](#)” on page 476.

A buyer creates an Item Not Received dispute by filing it on the eBay site. This generates an email to the seller, prompting the seller to respond to the dispute. The seller can respond on the eBay site or from an application using [AddDisputeResponse](#).

As a seller, you must respond within 10 days after the dispute was filed or risk action taken by eBay’s Trust and Safety team. When you respond to an using [AddDisputeResponse](#), you must specify a value for [DisputeActivity](#). The valid values are:

- **SellerOffersRefund** - if the seller offers the buyer a full refund if the item was not received or a partial refund if the item was significantly not as described.
- **SellerShippedItem** - if the seller wants to provide shipping information, for the original or replacement item.
- **SellerComment** - if the seller wants to communicate with the buyer.
- **SellerPaymentNotReceived** - if the seller has not received payment or the buyer’s payment has not cleared.

With any value of **DisputeActivity**, the seller is responding to the buyer. The buyer has the right to accept or reject the seller's response or close the dispute. In either case, the buyer uses the eBay site.

Each value of **DisputeActivity** can only be used with a certain value of **DisputeState**. You can check the current **DisputeState** by calling [GetDispute](#) or [GetUserDisputes](#). [Table 88-1](#) shows which values of **DisputeActivity** you can use for various **DisputeState** values.

Table 88-1 DisputeActivity and DisputeState in Item Not Received Disputes

When DisputeState Is	DisputeActivity Can Be
NotReceivedNoSellerResponse	SellerOffersRefund SellerShippedItem SellerComment SellerPaymentNotReceived
NotAsDescribedNoSellerResponse	SellerOffersRefund SellerComment
NotReceivedMutualCommunication	SellerOffersRefund SellerShippedItem SellerComment SellerPaymentNotReceived
NotAsDescribedMutualCommunication	SellerOffersRefund SellerComment

Each dispute is described by a **DisputeType** that contains all information about the dispute. For an Item Not Received dispute, the **DisputeType** contains **PurchaseProtection** and **Escalation** elements.

PurchaseProtection indicates whether the buyer is eligible for the eBay Purchase Protection Program (as described in the [eBay online help](#)). **Escalation** describes whether the dispute is eligible for escalation. For more details on purchase protection and escalation, see ["Buyer Disputes"](#) on page 476.

Using [AddDisputeResponse](#) for Item Not Received disputes triggers an **ItemNotReceived** notification. See ["INR \(ItemNotReceived\) Notifications"](#) on page 685.

AddDisputeResponse Input Fields

In the schema on which the API is based, the request to [AddDisputeResponse](#) takes a **DisputeID** and can take other values.

See the following document for information about the call's fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see ["Schema Reference Documentation"](#) on page 967.

AddDisputeResponse Output Fields

If the call to [AddDisputeResponse](#) is successful, it returns a call status of **Success**. If unsuccessful, it returns an error construct (see “[Error Handling](#)” on page 107).

See the following document for information about the call’s fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

Note that if a warning occurs, the result will contain both a call status and an error.

AddDisputeResponse Details

Prerequisites

- The call to [AddDisputeResponse](#) must be made by the seller who initiated an Unpaid Item dispute or is involved in an Item Not Received dispute.
- The dispute ID must be for a valid, open dispute.
- Be sure that you are taking action within the dispute time limits. For Unpaid Item disputes, see “[Closing an Unpaid Item Dispute](#)” on page 467. For Item Not Received disputes, see “[Buyer Disputes](#)” on page 476.
- The **DisputeActivity** must be valid for the current **DisputeState**. For Unpaid Item disputes, see “[DisputeState and DisputeActivity for Unpaid Items](#)” on page 473. For Item Not Received disputes, see [Table 88-1](#).
- If you pass **SellerAddInformation**, **SellerComment**, or **SellerPaymentNotReceived** for **DisputeActivity**, you must specify **MessageText**.

Working with the Response

[AddDisputeResponse](#) returns the current eBay time (in GMT) and a **CallStatus** if the response was executed successfully. Otherwise, [AddDisputeResponse](#) returns an error.

AddDisputeResponse Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

[AddDisputeResponse](#) returns only a call status and an eBay time.

Detail Levels

[AddDisputeResponse](#) does not support multiple detail levels. It is not necessary to specify a detail level for this call.

Pagination

[AddDisputeResponse](#) does not require pagination.

Data Filters

[AddDisputeResponse](#) does not provide data filters for reducing the volume of the response.

Reducing the Number of Calls You Make

Each call to [AddDisputeResponse](#) takes one action on a dispute and returns a call status. It is not necessary to reduce the number of calls you make to [AddDisputeResponse](#).

Testing AddDisputeResponse

- Set up at least one buyer, at least one seller, and at least one dispute in the Sandbox environment.
- Use the Sandbox seller auth token.
- Make a call to [GetDispute](#) or [GetUserDisputes](#) to check the dispute state before you call [AddDisputeResponse](#).
- Be sure that the action you are taking with [AddDisputeResponse](#) is allowed by the current dispute state. If you try to pass an action that is not allowed, you will see an error.
- Add a response, then test that you can retrieve the dispute with [GetDispute](#) and see the response in the dispute record.
- Close a dispute, then test that you can retrieve its state with [GetDispute](#) and that the dispute state shows the dispute is closed.
- Log in to the site as a buyer and create an Item Not Received dispute. Then, try to add a response as the seller.

AddDisputeResponse Samples

The samples in this section were created using the XML API. If you are working with the SOAP API, you can still use these samples as a guide, because they are equivalent to the structure in the body of a SOAP request and response.

Please note that these samples were created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if one of these samples does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in these samples.

Unpaid Item Dispute

[Example 88-1](#) adds a seller's comment to dispute ID 213833.

Example 88-1 Sample AddDisputeResponse Request

```
<?xml version="1.0" encoding="utf-8"?>
<AddDisputeResponseRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <SiteID>0</SiteID>
  <DisputeID>213833</DisputeID>
  <DisputeActivity>SellerAddInformation</DisputeActivity>
  <MessageText>Hello! I have tried in earnest to reach you. I will close this dispute in 3 days (Jan 7 if I do not hear from you.</MessageText>
</AddDisputeResponseRequest>
```

[Example 88-2](#) shows a successful response.

Example 88-2 Sample AddDisputeResponse Response

```
<?xml version="1.0" encoding="utf-8"?>
<AddDisputeResponseResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-03-10T18:11:46.449Z</Timestamp>
  <Ack>Success</Ack>
  <CorrelationID>
    00000000-00000000-00000000-00000000-00000000-00000000-0000000000
  </CorrelationID>
  <Version>467</Version>
  <Build>20050308050919</Build>
</AddDisputeResponseResponse>
```

The following example shows a seller closing a dispute with a **DisputeActivity** value of **SellerEndCommunication**. (This value can only be used in combination with certain values of **DisputeState**, as noted in [“DisputeState and DisputeActivity for Unpaid Items”](#) on page 473.)

Example 88-3 Closing a Dispute (Unified Schema XML API)

```
<?xml version="1.0" encoding="utf-8"?>
<AddDisputeResponseRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <DisputeID>213833</DisputeID>
  <DisputeActivity>SellerEndCommunication</DisputeActivity>
</AddDisputeResponseRequest>
```

Item Not Received Dispute

[Example 88-4](#) shows a request to **AddDisputeResponse** in which a seller adds shipping information to an Item Not Received dispute.

Example 88-4 Sample AddDisputeResponse Request

```
<?xml version="1.0" encoding="utf-8"?>
<AddDisputeResponseRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <SiteID>0</SiteID>
  <DisputeID>213818</DisputeID>
  <DisputeActivity>SellerShippedItem</DisputeActivity>
  <MessageText>Shipped to buyer today</MessageText>
```

Example 88-4 Sample AddDisputeResponse Request

```
<ShippingCarrierUsed>UPS</ShippingCarrierUsed>
<ShipmentTrackNumber>12345</ShipmentTrackNumber>
<ShippingTime>2005-03-10T21:43:35</ShippingTime>
</AddDisputeResponseRequest>
```

The response from the request is shown in [Example 88-5](#).

Example 88-5 Sample AddDisputeResponse Response

```
<?xml version="1.0" encoding="utf-8"?>
<AddDisputeResponseResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-03-10T21:47:55.011Z</Timestamp>
  <Ack>Success</Ack>
  <CorrelationID>
    00000000-00000000-00000000-00000000-00000000-0000000000
  </CorrelationID>
  <Version>401</Version>
  <Build>20050308050919</Build>
</AddDisputeResponseResponse>
```

Related Calls

- “[AddDispute](#)” on page 771
- “[GetDispute](#)” on page 828
- “ [GetUserDisputes](#)” on page 878
- “ [SellerReverseDispute](#)” on page 922

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 89

AddFixedPriceItem

Use this call to list a fixed-price item on eBay.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[AddFixedPriceItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 90

AddItem

Use this call to define a single new item and list it on a specified eBay site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[AddItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 91

AddItems

Use this call to define up to five (5) new items and list them on a specified eBay site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[AddItems](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 92

AddMemberMessageAAQToPartner

This call enables a buyer and seller in a transactional relationship to send messages to each other's My Messages Inboxes within 90 days of the creation of the transaction.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[AddMemberMessageAAQToPartner](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 93

AddMemberMessageRTQ

This call enables a seller to reply to a question about an active item listing.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[AddMemberMessageRTQ](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 94

AddMemberMessagesAAQToBidder

Sellers can use this call to contact both bidders and users who have made offers (via Best Offer) during an active listing.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[AddMemberMessagesAAQToBidder](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 95

AddOrder

Use this call to combine two or more transactions into a single order.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[AddOrder](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 96

AddSecondChanceItem

Use this call to extend a Second Chance Offer for an item to a specified non-winning bidder from the item's ended listing.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[AddSecondChanceOffer](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 97

AddToItemDescription

Use this call to add text to the end of the description section of an item listing, if there are 12 or more hours remaining in the auction.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[AddToItemDescription](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 98

AddToWatchList

Use this call to add one or more items to a user's My eBay watch list.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[AddToWatchList](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 99

AddTransactionConfirmationItem

On the the US eBay Motors site, use this call to create a new Transaction Confirmation Request (TCR) for the item. A TCR enables a buyer to purchase an item. Alternatively, this call can be used to verify that a new TCR can be created.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[AddTransactionConfirmationItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 100

CompleteSale

A seller can use this call to complete various end-of-sale tasks after an item transaction (or multiple-item order) has been created.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[CompleteSale](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 101

ConfirmIdentity

Use this call to return the ID of a user who has gone through an application's consent flow when getting an authorization token.

Note: The information for this call has been consolidated into the eBay Schema Input/Output Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[ConfirmIdentity](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 102

DeleteMyMessages

Use this call to remove selected messages for a given user.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[DeleteMyMessages](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 103

EndFixedPriceItem

Use this call to end a fixed-price item on eBay.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[EndFixedPriceItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 104

EndItem

Use this call to explicitly end an item listing before it would ordinarily end.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[EndItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 105

EndItems

Use this call to explicitly end up to ten (10) item listings before they would ordinarily end. .

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[EndItems](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 106

FetchToken

Use this call to retrieve an authentication token for a user.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[FetchToken](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 107

GetAccount

Use this call to retrieve a seller's account data.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetAccount](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 108

GetAdFormatLeads

Use this call to retrieve sales lead information for a Classified Ad listing.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetAdFormatLeads](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 109

GetAllBidders

Retrieves a list of the bidders for a specified listing.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetAllBidders](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 110

GetApiAccessRules

Use this call to retrieve the access rules for various calls and shows how many calls your application has made in the past day and hour.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetApiAccessRules](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 111

GetBestOffers

Use this call to retrieve the best offers associated with an [ItemID](#).

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetBestOffers](#)

[Input](#)

[Output](#)

[Samples](#)

tChapter 112

GetBidderList

Use this call to retrieve all items on which the user is currently bidding or which the buyer has won or purchased.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetBidderList](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 113

GetCategories

Use this call to retrieve a list of the categories for the specified eBay site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetCategories](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 114

GetCategory2CS

This call identifies categories that support Item Specifics and Pre-filled Item Information.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetCategory2CS](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 115

GetCategory2FinanceOffer

GetCategory2FinanceOffer has been deprecated. It is no longer functional as of July 2007.

Chapter 116

GetCategoryFeatures

Use this call to see what eBay features are defined for a site, and which categories support the feature.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetCategoryFeatures](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 117

GetCategoryListings

Retrieves a list of the items listed in a specified category.

This API call is being deprecated and will not be available after July 1, 2010. The functionality provided by this call is supported in the [Finding](#) API. If you're building new search applications, we recommend you start with the [Finding](#) API:

[Finding API](#)

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetCategoryListings](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 118

GetCategoryMappings

Use this call to retrieve a map of categories that are no longer active and the corresponding active categories on a given eBay site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetCategoryMappings](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 119

GetCategorySpecifics

Use this call to retrieve the the most popular custom Item Specific names and values for one or more leaf categories.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetCategorySpecifics](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 120

GetChallengeToken

Use this call to retrieve a botblock token and URLs for an image or audio clip that the user is to match.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetChallengeToken](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 121

GetCharities

Use [GetCharities](#) to search for nonprofit charity organizations that meet specified criteria.

This information can be used by 3rd-party developers to add a menu of potential nonprofit charity organizations that their sellers can donate profits to. The list of nonprofit charity organizations can be restricted by geographical region or mission area. Note eBay Giving Works may have minimum donation requirements for sellers. For more information, see ["Identifying Listings that Benefit Nonprofits"](#) on page 234.

- ["GetCharities Input Fields" on page 813](#)
- ["GetCharities Output Fields" on page 820](#)
- ["GetCharities Best Practices" on page 820](#)
- ["Testing GetCharities" on page 821](#)
- ["GetCharities Sample" on page 821](#)
- ["Related Calls" on page 822](#)
- ["Change History" on page 822](#)

GetCharities Input Fields

See the following document for information about the call's fields, including definitions and validation rules.



[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see ["Schema Reference Documentation" on page 967](#).

Valid CharityDomain Values

US and Motors

Table 121-1

CharityDomain Value	Domains (Mission Areas)
1	Arts and Humanities
2	Basic Needs and Human Services
3	Civic and Public Benefit
4	Education
5	Environment and Protection of Animals
6	Health and Sports

Table 121-1

CharityDomain Value	Domains (Mission Areas)
7	International
8	Religious

UK**Table 121-2**

CharityDomain Value	Domains (Mission Areas)
1	Culture, Sport, and Recreation
2	Education and Science
3	Health and Medicine
4	Social Services and Relief
5	Environment and Protection of Animals
6	Housing and Community Affairs
7	Civil Rights and Citizenship
8	Philanthropic Intermediation

Valid CharityRegion Values

US and Motors

Table 121-3

CharityRegion Value	Region Abbreviation	Region Name
1	DE	Delaware
2	AZ	Arizona
4	AK	Alaska
5	AR	Arkansas
7	CA	California
8	CO	Colorado
9	CT	Connecticut
10	AL	Alabama
11	DC	District of Columbia
12	FL	Florida
13	GA	Georgia
14	HI	Hawaii
15	ID	Idaho
16	IL	Illinois
17	IN	Indiana
18	IA	Iowa
19	KS	Kansas
20	KY	Kentucky
21	LA	Louisiana
22	ME	Maine
24	MD	Maryland
25	MA	Massachusetts
26	MI	Michigan
27	MN	Minnesota
28	MS	Mississippi
29	MO	Missouri
30	MT	Montana
31	NE	Nebraska
32	NV	Nevada
35	NH	New Hampshire
36	NJ	New Jersey
37	NM	New Mexico

Table 121-3

CharityRegion Value	Region Abbreviation	Region Name
38	NY	New York
39	NC	North Carolina
40	ND	North Dakota
42	OH	Ohio
43	OK	Oklahoma
45	OR	Oregon
46	PA	Pennsylvania
49	RI	Rhode Island
51	SC	South Carolina
52	SD	South Dakota
53	TN	Tennessee
54	TX	Texas
55	UT	Utah
56	VT	Vermont
57	VA	Virginia
58	WA	Washington
59	WV	West Virginia
60	WI	Wisconsin
61	WY	Wyoming
62	PR	Puerto Rico
63	VI	Virgin Islands
64	MP	Northern Mariana Islands
65	GU	Guam
66	AS	American Samoa
67	PW	Palau
68	FM	Federated States of Micronesia
69	MH	Marshall Islands

UK

Table 121-4

CharityRegion Value	Province/County
70	Avon
71	Bedfordshire
72	Berkshire
73	Bristol
74	Buckinghamshire
75	Cambridgeshire
76	Cheshire
77	Cleveland
78	Cornwall
79	Cumbria
80	Derbyshire
81	Devon
82	Dorset
83	Durham
84	East Riding of Yorkshire
85	East Sussex
86	Essex
87	Gloucestershire
88	Greater Manchester
89	Hampshire
90	Herefordshire
91	Hertfordshire
92	Humberside
93	Isle of Wight
94	Isles of Scilly
95	Kent
96	Lancashire
97	Leicestershire
98	Lincolnshire
99	London
100	Merseyside
101	Middlesex
102	Norfolk
103	North Yorkshire

Table 121-4

CharityRegion Value	Province/County
104	Northamptonshire
105	Northumberland
106	Nottinghamshire
107	Oxfordshire
108	Rutland
109	Shropshire
110	Somerset
111	South Yorkshire
112	Staffordshire
113	Suffolk
114	Surrey
115	Tyne and Wear
116	Warwickshire
117	West Midlands
118	West Sussex
119	West Yorkshire
120	Wiltshire
121	Worcestershire
122	Antrim
123	Armagh
124	Down
125	Fermanagh
126	Londonderry
127	Tyrone
128	default4
129	default5
130	Aberdeen City
131	Aberdeenshire
132	Angus
133	Argyll and Bute
134	Borders
135	Clackmannan
136	Dumfries and Galloway
137	Dundee (City of)
138	East Ayrshire

Table 121-4

CharityRegion Value	Province/County
139	East Dunbartonshire
140	East Lothian
141	East Renfrewshire
142	Edinburgh (City of)
143	Falkirk
144	Fife
145	Glasgow (City of)
146	Highland
147	Inverclyde
148	Midlothian
149	Moray
150	North Ayrshire
151	North Lanarkshire
152	Orkney
153	Perthshire and Kinross
154	Renfrewshire
155	Shetland
156	South Ayrshire
157	South Lanarkshire
158	Stirling
159	West Dunbartonshire
160	West Lothian
161	Western Isles
162	Blaenau Gwent
163	Bridgend
164	Caerphilly
165	Cardiff
166	Carmarthenshire
167	Ceredigion
168	Conwy
169	Denbighshire
170	Flintshire
171	Gwynedd
172	Isle of Anglesey
173	Merthyr Tydfil

Table 121-4

CharityRegion Value	Province/County
174	Monmouthshire
175	Neath Port Talbot
176	Newport
177	Pembrokeshire
178	Powys
179	Rhondda Cynon Taff
180	Swansea
181	Torfaen
182	The Vale of Glamorgan
183	Wrexham
184	Channel Islands
185	Isle of Man

GetCharities Output Fields

If the [GetCharities](#) execution is successful, it returns the requested nonprofit charity organization information. If unsuccessful, it returns an error construct (see “[Error Handling](#)” on page 107). Note that if a warning occurs, the result set will contain both normal data and an error construct.

Note: The **CharityID** value is returned as an **id** attribute in the parent **Charity** node. This information can be used in [AddItem](#) to add a Giving Works listing that benefits the charity specified by that **CharityID** value.

See the following document for information about the call’s fields, including definitions and validation rules.



[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

GetCharities Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

While it is possible to call [GetCharities](#) with no inputs to return all available nonprofit charity organizations, the large amount of data returned means that you should only do this when retrieving the initial list of nonprofits, and when refreshing the data to keep it current (see “[Keeping Data](#)

“Current” on page 821). See “[Data Filters](#)” on page 821 for recommendations on reducing the volume of the response.

Detail Levels

[GetCharities](#) does not support multiple detail levels. It is not necessary to specify a detail level for this call.

Pagination

[GetCharities](#) does not support pagination.

Data Filters

[GetCharities](#) does not provide data filters for reducing the volume of the response.

To minimize the number of nonprofit charity organizations returned, call [GetCharities](#) with a specified [CharityDomain](#) value. Make additional [GetCharities](#) calls with a different [CharityDomain](#) value if it is necessary to retrieve all available nonprofit charity organizations. See “[Valid CharityDomain Values](#)” on page 813 for a list of possible [CharityDomain](#) values.

Alternately, call [GetCharities](#) with a query string like [heart](#) specified in [CharityName](#) (to search title only) or [Query](#) (to search title and mission) to return only nonprofit charity organizations related to the query string.

Keeping Data Current

Because new nonprofits register with the eBay Giving Works providers every day, and, less frequently, charities become inactive, it is a best practice to ensure that your application can refresh the list of available nonprofits. Call [GetCharities](#) initially with no inputs to return all available nonprofit charity organizations. Then regularly call [GetCharities](#) and compare it to the cached data to ensure that the list is current.

Alternately, see “[Data Filters](#)” on page 821 for suggestions on retrieving complete data using multiple calls.

Testing GetCharities

- A certain number of charities are available in the sandbox for testing. Contact Developer Technical Support to create additional charities, or charities with special features.

GetCharities Sample

The following sample shows how to set a charity.

The sample in this section was created using the XML API. If you are working with the SOAP API, you can still use this sample as a guide, because it is equivalent to the structure in the body of a SOAP request and response.

Please note that this sample was created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if this sample does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in the sample.

This request was submitted to site ID 0:

Example 121-1 GetCharities Sample Input

```
<?xml version="1.0" encoding="utf-8"?>
<GetCharitiesRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <Version>425</Version>
    <RequesterCredentials>
        <eBayAuthToken>Your Token Here</eBayAuthToken>
    </RequesterCredentials>
    <Query>memory</Query>
</GetCharitiesRequest>
```

Here is the output from making that call:

Example 121-2 GetCharities Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<GetCharitiesResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2005-09-01T02:39:08.750Z</Timestamp>
    <Ack>Success</Ack>
    <CorrelationID>00000000-00000000-00000000-00000000-00000000-00000000-0000000000</CorrelationID>
    <Version>425</Version>
    <Build>e425_core_API_1718373_R1</Build>
    <Charity id="3002">
        <Name>Memorygift (JAX Foundation)</Name>
        <Mission>As a 501(c)(3) Non-Profit Foundation MemoryGift (JAX Foundation) will accept donations from the general public, sell hand made products and hold fund raisers to aid childrens charities.</Mission>
        <LogoURL>http://missionfish.ibs.aol.com/logos/MF2325.gif</LogoURL>
        <CharityRegion>0</CharityRegion>
    </Charity>
</GetCharitiesResponse>
```

Related Calls

[“AddItem” on page 782](#)

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 122

GetClientAlertsAuthToken

This call retrieves a Client Alerts authentication token for a user.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetClientAlertsAuthToken](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 123

GetContextualKeywords

Use this call to retrieve top-ranked contextual eBay keywords and categories for a specified web page.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetContextualKeywords](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 124

GetCrossPromotions

Use this call to retrieve a list of upsell or cross-sell items for one referring item.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetCrossPromotions](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 125

GetDescriptionTemplates

Use this call to retrieve the Theme and Layout specifications for the display of an item's description.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetDescriptionTemplates](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 126

GetDispute

Returns the details of a specific Unpaid Item or Item Not Received dispute.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetDispute](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 127

GeteBayDetails

Use this call to retrieve the available meta-data for the specified eBay site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GeteBayDetails](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 128

GeteBayOfficialTime

Use this call to determine the eBay system time.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GeteBayOfficialTime](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 129

GetFeedback

Retrieves the feedback data for a specified user, transaction, or item.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetFeedback](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 130

GetFinanceOffers

GetFinanceOffers has been deprecated. It is no longer functional as of July 2007.

Chapter 131

GetItem

Use this call to retrieve the data for a single item listed on an eBay site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 132

GetItemsAwaitingFeedback

Use this call to return transactions in which the user was involved and for which feedback is still needed from either the buyer or seller.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetItemsAwaitingFeedback](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 133

GetItemShipping

This call is analogous to the Shipping Calculator seen in both the buyer and seller web pages—it allows a buyer or seller to obtain shipping cost information for an existing item.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetItemShipping](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 134

GetItemTransactions

Retrieves transaction information for a specified item ID.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetItemTransactions](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 135

GetMemberMessages

Use this call to retrieve a list of the messages buyers have posted about your active item listings.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetMemberMessages](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 136

GetMessagePreferences

Use this call to retrieve a seller's Ask Seller a Question (ASQ) subjects.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetMessagePreferences](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 137

GetMyeBayBuying

This call returns items from the user's My eBay list, including Items I'm Watching, Items I'm Bidding On, Items I Made Best Offers On, Items I've Won, Items I Didn't Win, My Favorite Searches, and My Favorite Sellers.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetMyeBayBuying](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 138

GetMyeBayReminders

This call retrieves reminders displayed for the user from the user's My eBay account.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetMyeBayReminders](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 139

GetMyeBaySelling

Use this call to return information from the All Selling section of the authenticated user's My eBay account.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetMyeBaySelling](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 140

GetMyMessages

Use this call to gather information about the alerts and threaded messages sent to a given user's My Messages mailbox.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetMyMessages](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 141

GetNotificationPreferences

[GetNotificationPreferences](#) is part of eBay Platform Notifications, an optional feature that alerts applications when certain events occur on the eBay site.

Before using this call, please review the topic “[Working with Platform Notifications](#)” on page 623 to understand the requirements and limitations of the Platform Notifications feature.

Use [GetNotificationPreferences](#) to determine what notification preferences have been set. These preferences can be at the application level or the user level.

- “[GetNotificationPreferences Input Fields](#)” on page 843
- “[GetNotificationPreferences Output Fields](#)” on page 844
- “[GetNotificationPreferences Details](#)” on page 844
- “[GetNotificationPreferences Best Practices](#)” on page 844
- “[Testing GetNotificationPreferences](#)” on page 845
- “[GetNotificationPreferences Samples](#)” on page 845
- “[Related Calls](#)” on page 848
- “[Change History](#)” on page 848

GetNotificationPreferences Input Fields

[GetNotificationPreferences](#) takes as input a **NotificationRoleCodeType** value, which determines the preference type to return. The table below shows the input field and the type for the [GetNotificationPreferences](#) request object.

Table 141-1

Input Field	Required?	Type/Code List	Purpose
PreferenceLevel	Y	NotificationRoleCodeType	Determines which preference type to return, either as application or user-based data.

See the following document for information about the call’s fields, including definitions and validation rules.



[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

GetNotificationPreferences Output Fields

If the [GetNotificationPreferences](#) execution is successful, it will return the notifications set for a particular event or a group of events, depending on what preference types were selected using the [SetNotificationPreferences](#). This allows the user to see application or user (depending on the boolean value(s) selected for input) notification settings from the eBay site.

If unsuccessful, the call will generate error message(s). To see the list of error messages that can be returned, see “[Error Message Strings](#)” on page 966.

Table 141-2

Return Fields	Type/Code List	Purpose
ApplicationDeliveryPreferences	ApplicationDeliveryPreferencesType	Specifies application-based event preferences that have been enabled.
UserDeliveryPreferenceArray	NotificationEnableArrayType	Specifies user-based event preferences that have been enabled.

See the following document for information about the call’s fields, including definitions and validation rules.



[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

GetNotificationPreferences Details

Prerequisites

- The requestor must be a registered eBay user.

Working with the Response

[GetNotificationPreferences](#) gets specific event notifications information about either application or user-based events, depending on the boolean values selected in the input.

GetNotificationPreferences Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

For [GetNotificationPreferences](#), this is not applicable.

Detail Levels

[GetNotificationPreferences](#) does not support multiple detail levels.

Pagination

[GetNotificationPreferences](#) does not require or support pagination.

Data Filters

[GetNotificationPreferences](#) does not provide data filters for reducing the volume of the response.

Reducing the Number of Calls You Make

- The [GetNotificationPreferences](#) is used to get specific event types, so reducing the number of calls is not applicable.

Testing GetNotificationPreferences

You can test [GetNotificationPreferences](#) in the Sandbox. The preferences that you set in the Sandbox do not apply to Production, so once you have tested the call, you can use it to set your preferences for the Production environment.

GetNotificationPreferences Samples

The following sample shows an example of a [GetNotificationPreferences](#) request with a [PreferenceLevel](#) value of [User](#).

The samples in this section were created using the XML API. If you are working with the SOAP API, you can still use these samples as a guide, because they are equivalent to the structure in the body of a SOAP request and response.

Please note that these samples were created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if one of these samples does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in these samples.

Example 141-1 GetNotificationPreferences Sample Input for User

```
<?xml version="1.0" encoding="utf-8"?>
<RequesterCredentials>
    <eBayAuthToken>Your Token Here</eBayAuthToken>
</RequesterCredentials>
<GetNotificationPreferencesRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <Version>423</Version>
    <PreferenceLevel>User</PreferenceLevel>
</GetNotificationPreferencesRequest>
```

This sample show the output from that call

Example 141-2 GetNotificationPreferences Sample Output for User

```
<?xml version="1.0" encoding="UTF-8"?>
<GetNotificationPreferencesResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-08-24T02:19:01.945Z</Timestamp>
  <Ack>Success</Ack>
  <CorrelationID>00000000-00000000-00000000-00000000-00000000-00000000-0000000000</
CorrelationID>
  <Version>423</Version>
  <Build>e423_core_API_1685121_R1</Build>
  <UserDeliveryPreferenceArray>
    <NotificationEnable>
      <EventType>OutBid</EventType>
      <EventEnable>Enable</EventEnable>
    </NotificationEnable>
    <NotificationEnable>
      <EventType>EndOfAuction</EventType>
      <EventEnable>Enable</EventEnable>
    </NotificationEnable>
    <NotificationEnable>
      <EventType>FixedPriceTransaction</EventType>
      <EventEnable>Enable</EventEnable>
    </NotificationEnable>
    <NotificationEnable>
      <EventType>Feedback</EventType>
      <EventEnable>Enable</EventEnable>
    </NotificationEnable>
    <NotificationEnable>
      <EventType>SecondChanceOffer</EventType>
      <EventEnable>Enable</EventEnable>
    </NotificationEnable>
    <NotificationEnable>
      <EventType>BuyerResponseDispute</EventType>
      <EventEnable>Enable</EventEnable>
    </NotificationEnable>
    <NotificationEnable>
      <EventType>SellerOpenedDispute</EventType>
      <EventEnable>Enable</EventEnable>
    </NotificationEnable>
    <NotificationEnable>
      <EventType>SellerRespondedToDispute</EventType>
      <EventEnable>Enable</EventEnable>
    </NotificationEnable>
    <NotificationEnable>
      <EventType>SellerClosedDispute</EventType>
      <EventEnable>Enable</EventEnable>
    </NotificationEnable>
    <NotificationEnable>
      <EventType>BestOffer</EventType>
      <EventEnable>Enable</EventEnable>
    </NotificationEnable>
```

Example 141-2 GetNotificationPreferences Sample Output for User

```

<NotificationEnable>
  <EventType>ItemRevised</EventType>
  <EventEnable>Enable</EventEnable>
</NotificationEnable>
<NotificationEnable>
  <EventType>MyMessagesAlert</EventType>
  <EventEnable>Enable</EventEnable>
</NotificationEnable>
<NotificationEnable>
  <EventType>MyMessageseBayMessage</EventType>
  <EventEnable>Enable</EventEnable>
</NotificationEnable>
<NotificationEnable>
  <EventType>MyMessagesM2MMessage</EventType>
  <EventEnable>Enable</EventEnable>
</NotificationEnable>
</UserDeliveryPreferenceArray>
</GetNotificationPreferencesResponse>

```

The following sample shows an example of a GetNotificationPreferences request with a **PreferenceLevel** value of **Application**.

Example 141-3 GetNotificationPreferences Sample Input for Application

```

<?xml version="1.0" encoding="utf-8"?>
<RequesterCredentials>
  <eBayAuthToken>Your Token Here</eBayAuthToken>
</RequesterCredentials>
<GetNotificationPreferencesRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <Version>423</Version>
  <PreferenceLevel>Application</PreferenceLevel>
</GetNotificationPreferencesRequest>

```

This sample show the output from that call

Example 141-4 GetNotificationPreferences Sample Output for Application

```

<?xml version="1.0" encoding="UTF-8"?>
<GetNotificationPreferencesResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-08-24T02:28:04.430Z</Timestamp>
  <Ack>Success</Ack>
  <CorrelationID>00000000-00000000-00000000-00000000-00000000</CorrelationID>
  <Version>423</Version>
  <Build>e423_core_API_1685121_R1</Build>
  <ApplicationDeliveryPreferences>
    <ApplicationURL>http://www.ebay.com</ApplicationURL>
    <ApplicationEnable>Enable</ApplicationEnable>
    <NotificationPayloadType>eBLSchemaSOAP</NotificationPayloadType>
  </ApplicationDeliveryPreferences>
</GetNotificationPreferencesResponse>

```

Related Calls

[“SetNotificationPreferences” on page 929](#)
[“GetNotificationsUsage” on page 849](#)

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 142

GetNotificationsUsage

GetNotificationsUsage retrieves information about platform notifications sent to a given application. Use **GetNotificationsUsage** to confirm you are being sent all the platform notifications to which you have subscribed or to troubleshoot problems with notification delivery.

[AddItem](#)

[Input](#)

[Output](#)

[Samples](#)

GetNotificationsUsage Samples

GetNotificationsUsage can be used with or without specifying an item ID.

The samples in this section were created using the XML API. If you are working with the SOAP API, you can still use these samples as a guide, because they are equivalent to the structure in the body of a SOAP request and response.

Please note that these samples were created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if one of these samples does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in these samples.

[Example 142-3](#) requests general notification activity for your application for a three day period.

Example 142-1 GetNotificationsUsage Sample Input

```
<?xml version="1.0" encoding="UTF-8"?>
<GetNotificationsUsageRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>YOUR TOKEN HERE</eBayAuthToken>
  </RequesterCredentials>
  <Version>439</Version>
  <StartTime>2005-12-07T21:00:00.000Z</StartTime>
  <EndTime>2005-12-09T20:59:00.000Z</EndTime>
</GetNotificationsUsageRequest>
```

Example 142-4 shows the response, with some mark-down and mark-up activity.

Example 142-2 GetNotificationsUsage Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<GetNotificationsUsageResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-12-09T21:00:44.540Z</Timestamp>
  <Ack>Success</Ack>
  <Version>439</Version>
  <Build>e439_core_API_2160775_R1</Build>
  <StartTime>2005-12-07T21:00:00.000Z</StartTime>
  <EndTime>2005-12-09T20:59:00.000Z</EndTime>
  <MarkUpMarkDownHistory>
    <MarkUpMarkDownEvent>
      <Type>MarkDown</Type>
      <Time>2005-12-08T03:10:06.000Z</Time>
      <Reason>Automatically marked down</Reason>
    </MarkUpMarkDownEvent>
    <MarkUpMarkDownEvent>
      <Type>MarkUp</Type>
      <Time>2005-12-08T05:04:37.000Z</Time>
    </MarkUpMarkDownEvent>
  </MarkUpMarkDownHistory>
  <NotificationStatistics>
    <DeliveredCount>1002</DeliveredCount>
    <QueuedNewCount>0</QueuedNewCount>
    <QueuedPendingCount>0</QueuedPendingCount>
    <ExpiredCount>1184</ExpiredCount>
    <ErrorCount>212</ErrorCount>
  </NotificationStatistics>
</GetNotificationsUsageResponse>
```

Example 142-3 requests notification activity for a specific item for a 24-hour period.

Example 142-3 GetNotificationsUsage Sample Input

```
<?xml version="1.0" encoding="UTF-8"?>
<GetNotificationsUsageRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>YOUR TOKEN HERE</eBayAuthToken>
  </RequesterCredentials>
  <Version>439</Version>
  <StartTime>2005-12-08T21:00:00.000Z</StartTime>
  <EndTime>2005-12-09T21:00:00.000Z</EndTime>
  <ItemID>4076208069</ItemID>
</GetNotificationsUsageRequest>
```

Example 142-4 shows the response, with notification details.

Example 142-4 GetNotificationsUsage Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<GetNotificationsUsageResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-12-09T21:00:44.540Z</Timestamp>
  <Ack>Success</Ack>
  <Version>439</Version>
  <Build>e439_core_API_2160775_R1</Build>
  <StartTime>2005-12-12T23:51:19.345Z</StartTime>
  <EndTime>2005-12-13T23:51:19.345Z</EndTime>
```

Example 142-4 GetNotificationsUsage Sample Output

```
<NotificationDetailsArray>
  <NotificationDetails>
    <DeliveryURL>http://www.yourapp.com/eBayNotification/eBayNotification.aspx
  </DeliveryURL>
    <ReferenceID>4076208069:178713</ReferenceID>
    <ExpirationTime>2005-12-08T00:17:58.000Z</ExpirationTime>
    <Type>EndOfAuction</Type>
    <Retries>1</Retries>
    <DeliveryStatus>Delivered</DeliveryStatus>
    <NextRetryTime>2005-12-07T02:17:58.000Z</NextRetryTime>
    <DeliveryTime>2005-12-07T00:18:01.000Z</DeliveryTime>
    <ErrorMessage>|[Notification HttpEventSender: URL http://www.yourapp.com/eBayNotification/
eBayNotification.aspx: HTTP Client returned 'Success' (HTTP status: 200, 'OK')]</ErrorMessage>
  </NotificationDetails>
  <NotificationDetails>
    <DeliveryURL>http://www.yourapp.com/eBayNotification/eBayNotification.aspx
  </DeliveryURL>
    <ReferenceID>4076208069:20872261</ReferenceID>
    <ExpirationTime>2005-12-08T00:17:58.000Z</ExpirationTime>
    <Type>EndOfAuction</Type>
    <Retries>1</Retries>
    <DeliveryStatus>Delivered</DeliveryStatus>
    <NextRetryTime>2005-12-07T02:17:58.000Z</NextRetryTime>
    <DeliveryTime>2005-12-07T00:18:01.000Z</DeliveryTime>
    <ErrorMessage>|[Notification HttpEventSender: URL http://www.yourapp.com/eBayNotification/
eBayNotification.aspx: HTTP Client returned 'Success' (HTTP status: 200, 'OK')]</ErrorMessage>
  </NotificationDetails>
</NotificationDetailsArray>
<MarkUpMarkDownHistory/>
<NotificationStatistics>
  <DeliveredCount>157</DeliveredCount>
  <QueuedNewCount>0</QueuedNewCount>
  <QueuedPendingCount>0</QueuedPendingCount>
  <ExpiredCount>0</ExpiredCount>
  <ErrorCount>136</ErrorCount>
</NotificationStatistics>
</GetNotificationsUsageResponse>
```

Related Calls

[“GetNotificationPreferences” on page 843](#)
[“SetNotificationPreferences” on page 929](#)

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 143

GetOrders

Use this call to retrieve all of the orders for which the user is a participant (as either buyer or seller) that meet the criteria specified in the request object.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetOrders](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 144

GetOrderTransactions

Use this call to retrieve information about one or more orders, or one or more transactions (or both).

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetOrderTransactions](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 145

GetPromotionalSaleDetails

Use this call to obtain information about promotional sales, which are discounts that apply to item listings.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetPromotionalSaleDetails](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 146

GetPromotionRules

This call retrieves the promotion rules that are associated with a specific item or store category.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetPromotionRules](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 147

GetSearchResults

Use this call to search for item listings based on keywords and other criteria that you define.

Note: This API call is being deprecated and will not be available after July 2010. The functionality provided by this call is supported in the Finding API. If you're building new search applications, we recommend you start with the Finding API:
[Finding API](#)

The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide. See the following sections in the Call Reference for information about this call:

[GetSearchResults](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 148

GetSellerDashboard

Use this call to retrieve a brief summary of the requester's status as an eBay seller. .

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetSellerDashboard](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 149

GetSellerEvents

Retrieves a list of the items a specified seller has listed.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetSellerEvents](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 150

GetSellerList

Use this call to get a list of the current active items posted by the authenticated user, including the related item data.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetSellerList](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 151

GetSellerPayments

Use [GetSellerPayments](#) to retrieve a summary of pending or paid payments that Half.com created for the seller identified by the authentication token in the request.

The call retrieves payments that occurred within a particular pay period.

Each payment is for one transaction for one item in one order. An order can contain transactions for multiple items from multiple sellers, but this call only retrieves payments that are relevant to one seller. For basic information about Half.com listings and payments, see “[Half.com](#)” on page 574.

Important: The Sandbox does not support Half.com.

- “[GetSellerPayments Input Fields](#)” on page 860
- “[GetSellerPayments Output Fields](#)” on page 860
- “[GetSellerPayments Details](#)” on page 861
- “[GetSellerPayments Best Practices](#)” on page 861
- “[GetSellerPayments Samples](#)” on page 862
- “[Related Calls](#)” on page 863
- “[Change History](#)” on page 863

GetSellerPayments Input Fields

[GetSellerPayments](#) takes a time range and payment status (pending or paid) as input.

See the following document for information about the call’s fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

GetSellerPayments Output Fields

If the execution of [GetSellerPayments](#) is successful, it retrieves information about pending or paid payments from Half.com to the seller. If unsuccessful, the call returns an error construct (see “[Error Handling](#)” on page 107). Note that if a warning occurs, the result set will contain both normal data and an error construct.

See the following document for information about the call's fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

GetSellerPayments Details

Prerequisites

See “[Working with Half.com Listings](#)” on page 558 for requirements that sellers need to meet in order to retrieve data using [GetSellerPayments](#).

Working with the Response

[GetSellerPayments](#) returns data for pending or paid payments into properties of the [GetSellerPayments](#) response object.

GetSellerPayments Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

[GetSellerPayments](#) retrieves a maximum of 200 payments per call. It only retrieves payments for one pay period at a time.

Detail Levels

[GetSellerPayments](#) does not support multiple detail levels. It is not necessary to specify a detail level for this call.

Pagination

[GetSellerPayments](#) supports pagination.

Data Filters

[GetSellerPayments](#) provides a payment time filter for identifying the pay period of interest. However, using this filter does not necessarily reduce the volume of the response.

Reducing the Number of Calls You Make

There are no special features to reduce the number of times you execute [GetSellerPayments](#).

GetSellerPayments Samples

The sample in this section was created using the XML API. If you are working with the SOAP API, you can still use this sample as a guide, because it is equivalent to the structure in the body of a SOAP request and response.

Please note that this sample was created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if this sample does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in the sample.

The following sample shows a request for pending listings and the resulting response. This request was submitted to site ID 0 (the US site).

Example 151-1 GetSellerPayments Sample Input

```
<?xml version="1.0" encoding="utf-8"?>
<GetSellerPaymentsRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <Version>431</Version>
  <RequesterCredentials>
    <eBayAuthToken>YOURTOKENHERE</eBayAuthToken>
  </RequesterCredentials>
  <PaymentStatus>Pending</PaymentStatus>
  <PaymentTimeFrom>2005-10-01T00:00:00.000Z</PaymentTimeFrom>
  <PaymentTimeTo>2005-10-12T12:00:00.000Z</PaymentTimeTo>
  <Pagination>
    <EntriesPerPage>2</EntriesPerPage>
  </Pagination>
</GetSellerPaymentsRequest>
```

Here is the output from making that call:

Example 151-2 GetSellerPayments Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<GetSellerPaymentsResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-10-20T06:38:24.086Z</Timestamp>
  <Ack>Success</Ack>
  <Version>431</Version>
  <Build>e431_core_API_1926042_R1</Build>
  <PaginationResult>
    <TotalNumberOfPages>3</TotalNumberOfPages>
    <TotalNumberOfEntries>5</TotalNumberOfEntries>
  </PaginationResult>
  <HasMorePayments>true</HasMorePayments>
```

Example 151-2 GetSellerPayments Sample Output

```
<SellerPayment>
  <ItemID>14014827713</ItemID>
  <TransactionID>35171</TransactionID>
  <OrderID>0</OrderID>
  <SellerInventoryID>0025403400</SellerInventoryID>
  <ExternalProductID>
    <Value>0025403400</Value>
    <Type>ISBN</Type>
  </ExternalProductID>
  <Title>Flying the Commercial Flight Test (Hard)</Title>
  <TransactionPrice currencyID="USD">13.0</TransactionPrice>
  <ShippingReimbursement currencyID="USD">2.33</ShippingReimbursement>
  <Commission currencyID="USD">1.95</Commission>
  <AmountPaid currencyID="USD">13.38</AmountPaid>
  <PaidTime>2005-10-13T22:04:22.000Z</PaidTime>
</SellerPayment>
<SellerPayment>
  <ItemID>14014827116</ItemID>
  <TransactionID>34910</TransactionID>
  <OrderID>0</OrderID>
  <SellerInventoryID>0066238269</SellerInventoryID>
  <ExternalProductID>
    <Value>0066238269</Value>
    <Type>ISBN</Type>
  </ExternalProductID>
  <Title>The Magician's Nephew (Hard)</Title>
  <TransactionPrice currencyID="USD">31.0</TransactionPrice>
  <ShippingReimbursement currencyID="USD">2.33</ShippingReimbursement>
  <Commission currencyID="USD">4.65</Commission>
  <AmountPaid currencyID="USD">28.68</AmountPaid>
  <PaidTime>2005-10-12T22:35:27.000Z</PaidTime>
</SellerPayment>
<PaymentsPerPage>2</PaymentsPerPage>
<PageNumber>1</PageNumber>
<ReturnedPaymentCountActual>2</ReturnedPaymentCountActual>
</GetSellerPaymentsResponse>
```

Related Calls

[“IssueRefund” on page 900](#)

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 152

GetSellerTransactions

Retrieves transaction information for the user for which the call is made.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetSellerTransactions](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 153

GetSessionID

Use this call to retrieve a **SessionID** for a user.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetSessionID](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 154

GetShippingDiscountProfiles

Use this call to retrieve the shipping discount profiles previously defined by the user and to see the current settings for combined packaging and handling costs.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetShippingDiscountProfiles](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 155

GetStore

Use this call to retrieve a seller's eBay store configuration.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetStore](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 156

GetStoreCategoryUpdateStatus

Use this call to retrieve the status of the processing progress for category structure changes specified earlier with [SetStoreCategories](#).

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetStoreCategoryUpdateStatus](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 157

GetStoreCustomPage

Use this call to retrieve the custom pages for a user's eBay Store.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetStoreCustomPage](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 158

GetStoreOptions

Use this call to retrieve the list of current eBay Stores options.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetStoreOptions](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 159

GetStorePreferences

Use this call to retrieve the preferences for a user's eBay Store.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetUser](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 160

GetSuggestedCategories

Returns a list of categories with the highest number of items whose titles contain keywords that you specify.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetSuggestedCategories](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 161

GetTaxTable

This call retrieves the tax table for a user on a given site, or it retrieves the valid jurisdictions (if any) for a given site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetTaxTable](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 162

GetTokenStatus

Use this call to retrieve the status of a user token, which represents permission for an application to access, on the user's behalf, eBay data using eBay APIs.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetTokenStatus](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 163

GetUser

Use this call to retrieve the data for a single user registered on an eBay site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetUser](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 164

GetUserContactDetails

Use this call to return contact information for a specified user, given that a bidding relationship (as either a buyer or seller) exists between the caller and the user. .

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetUserContactDetails](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 165

GetUserDisputes

Retrieves a list of all disputes involving the requester as buyer or seller, including both Unpaid Item and Item Not Received disputes.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetUserDisputes](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 166

GetUserPreferences

Use this call to retrieve the user preferences for a single user registered on an eBay site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[GetUserPreferences](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 167

GetVeROReasonCodeDetails

Use [GetVeROReasonCodeDetails](#) to retrieve a list of the valid reason codes for one or all eBay sites. When an item is reported with [VeROReportItems](#) for alleged infringement, you must specify a valid reason code.

- “[GetVeROReasonCodeDetails Input Fields](#)” on page 880
- “[GetVeROReasonCodeDetails Output Fields](#)” on page 880
- “[GetVeROReasonCodeDetails Sample](#)” on page 880
- “[GetVeROReasonCodeDetails Details](#)” on page 882
- “[GetVeROReasonCodeDetails Best Practices](#)” on page 882
- “[Testing GetVeROReasonCodeDetails](#)” on page 883
- “[Related Calls](#)” on page 883
- “[Change History](#)” on page 883

GetVeROReasonCodeDetails Input Fields

See the following document for information about the call’s fields, including definitions and validation rules.



[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

GetVeROReasonCodeDetails Output Fields

See the following document for information about the call’s fields, including definitions and validation rules.



[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

GetVeROReasonCodeDetails Sample

The sample in this section was created using the XML API. If you are working with the SOAP API, you can still use this sample as a guide, because it is equivalent to the structure in the body of a SOAP request and response.

Please note that this sample was created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if this sample does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in the sample.

The following call sample retrieves VeRO reason codes for all sites.

Example 167-1 GetVeROReasonCodeDetails Sample Input

```
<?xml version="1.0" encoding="utf-8"?>
<GetVeROReasonCodeDetailsRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>YOUR TOKEN HERE</eBayAuthToken>
  </RequesterCredentials>
  <Version>491</Version>
  <ReturnAllSites>true</ReturnAllSites>
</GetVeROReasonCodeDetailsRequest>
```

Here is the output from making that call:

Example 167-2 GetVeROReasonCodeDetails Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<GetVeROReasonCodeDetailsResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-02-02T23:56:18.840Z</Timestamp>
  <Ack>Success</Ack>
  <Version>491</Version>
  <Build>e491_core_Bundled_3958874_R1</Build>
  <VeROReasonCodeDetails>
    <VeROSiteDetail>
      <Site>US</Site>
      <ReasonCodeDetail codeID="9035">
        <BriefText>Trademark--Item Infringement</BriefText>
        <DetailedText>Trademark owner doesn't make this type of product</DetailedText>
      </ReasonCodeDetail>
      <ReasonCodeDetail codeID="9036">
        <BriefText>Trademark--Item Infringement</BriefText>
        <DetailedText>Item(s) is unlawful replica of a product made by the trademark owner</DetailedText>
      </ReasonCodeDetail>
      <ReasonCodeDetail codeID="9037">
        <BriefText>Trademark--Item Infringement</BriefText>
        <DetailedText>Item(s) is unlawful importation of product bearing trademark</DetailedText>
      </ReasonCodeDetail>
      ... additional reason code details returned ...
      <ReasonCodeDetail codeID="9052">
        <BriefText>Other Infringement</BriefText>
        <DetailedText>Other</DetailedText>
      </ReasonCodeDetail>
    </VeROSiteDetail>
```

Example 167-2 GetVeROReasonCodeDetails Sample Output

```
<VeROSiteDetail>
  <Site>Canada</Site>
  <ReasonCodeDetail codeID="9035">
    <BriefText>Trademark--Item Infringement</BriefText>
    <DetailedText>Trademark owner doesn't make this type of product</DetailedText>
  </ReasonCodeDetail>
  ... additional reason code details returned ...
  <ReasonCodeDetail codeID="9051">
    <BriefText>Other Infringement</BriefText>
    <DetailedText>Listing(s) content violates a celebrity's right of publicity</
DetailedText>
  </ReasonCodeDetail>
</VeROSiteDetail>
... additional site details returned ...
<VeROSiteDetail>
  <Site>Ireland</Site>
  <ReasonCodeDetail codeID="9035">
    <BriefText>Trademark--Infringing Item</BriefText>
    <DetailedText>Trade mark owner doesn't make this type of product</DetailedText>
  </ReasonCodeDetail>
  ... additional reason code details returned ...
  <ReasonCodeDetail codeID="9051">
    <BriefText>Other Infringement</BriefText>
    <DetailedText>Listing(s) content violates a celebrity's right of publicity</
DetailedText>
  </ReasonCodeDetail>
</VeROSiteDetail>
</VeROReasonCodeDetails>
</GetVeROReasonCodeDetailsResponse>
```

GetVeROReasonCodeDetails Details

Prerequisites

- The requestor must be a registered eBay user.
- The requestor must be a member of the Verified Rights Owner (VeRO) Program to use this call.

GetVeROReasonCodeDetails Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

There is currently no limit on number of records returned.

Detail Levels

GetVeROReasonCodeDetails does not support multiple detail levels. It is not necessary to specify a detail level for this call.

Pagination

`GetVeROReasonCodeDetails` does not require pagination, but you can

Data Filters

`GetVeROReasonCodeDetails` does not provide data filters for reducing the volume of the response.

Testing GetVeROReasonCodeDetails

- Testing for VeRO calls is not supported in the sandbox environment. Contact Developer Support for information on how to test in the production environment.

Related Calls

See “[GetVeROReportStatus](#)” on page 884.

See “[VeROReportItems](#)” on page 948.

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 168

GetVeROReportStatus

Use [GetVeROReportStatusCall](#) to retrieve the VeRO status for items you have reported for infringement. You can retrieve the status of individual items you have reported or, by specifying VeROReportPacketID, you can retrieve status for all items reported with a given [VeROReportItems](#) request. You can also retrieve items that were reported during a given time period. If no input parameters are specified, status is returned on all items you have reported in the last two years.

You must be a member of the Verified Rights Owner (VeRO) Program to use this call.

- ["GetVeROReportStatus Input Fields" on page 884](#)
- ["GetVeROReportStatus Output Fields" on page 884](#)
- ["GetVeROReportStatus Samples" on page 885](#)
- ["GetVeROReportStatus Details" on page 887](#)
- ["GetVeROReportStatus Best Practices" on page 887](#)
- ["Testing GetVeROReportStatus" on page 887](#)
- ["Related Calls" on page 888](#)
- ["Change History" on page 888](#)

GetVeROReportStatus Input Fields

See the following document for information about the call's fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see ["Schema Reference Documentation" on page 967](#).

GetVeROReportStatus Output Fields

See the following document for information about the call's fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see ["Schema Reference Documentation" on page 967](#).

GetVeROReportStatus Samples

The samples in this section were created using the XML API. If you are working with the SOAP API, you can still use these samples as a guide, because they are equivalent to the structure in the body of a SOAP request and response.

Please note that these samples were created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if one of these samples does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in these samples.

The following call sample requests the status for a specific packet of VeRO reported items submitted with a [VeROReportItems](#) request.

Example 168-1 GetVeROReportStatus Sample Input (packet status)

```
<?xml version="1.0" encoding="utf-8"?>
<GetVeROReportStatusRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>YOUR TOKEN HERE</eBayAuthToken>
  </RequesterCredentials>
  <VeROReportPacketID>5000051334</VeROReportPacketID>
</GetVeROReportStatusRequest>
```

Here is the output from making that call:

Example 168-2 GetVeROReportStatus Sample Output (packet status)

```
<?xml version="1.0" encoding="utf-8"?>
<GetVeROReportStatusResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-02-02T23:56:18.840Z</Timestamp>
  <Ack>Success</Ack>
  <Version>491</Version>
  <Build>e491_core_Bundled_3958874_R1</Build>
  <VeROReportPacketID>5000051334</VeROReportPacketID>
  <VeROReportPacketStatus>Processed</VeROReportPacketStatus>
</GetVeROReportStatusResponse>
```

The following call sample requests status for all VeRO reported items submitted (by any means) during the specified time period.

Example 168-3 GetVeROReportStatus Sample Input

```
<?xml version="1.0" encoding="utf-8"?>
<GetVeROReportStatusRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <RequesterCredentials>
        <eBayAuthToken>YOUR TOKEN HERE</eBayAuthToken>
    </RequesterCredentials>
    <TimeFrom>2006-11-01 00:12:07</TimeFrom>
    <TimeTo>2006-12-01 00:12:07</TimeTo>
    <Pagination>
        <EntriesPerPage>50</EntriesPerPage>
        <PageNumber>1</PageNumber>
    </Pagination>
</GetVeROReportStatusRequest>
```

Here is the output from making that call:

Example 168-4 GetVeROReportStatus Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<GetVeROReportStatusResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2005-02-02T23:56:18.840Z</Timestamp>
    <Ack>Success</Ack>
    <Version>491</Version>
    <Build>e491_core_Bundled_3964543_R1</Build>
    <PaginationResult>
        <TotalNumberOfPages>7</TotalNumberOfPages>
        <TotalNumberOfEntries>326</TotalNumberOfEntries>
    </PaginationResult>
    <ReportedItemDetails>
        <ReportedItem>
            <ItemID>330000432935</ItemID>
            <ItemStatus>Received</ItemStatus>
        </ReportedItem>
        <ReportedItem>
            <ItemID>150000349786</ItemID>
            <ItemStatus>Submitted</ItemStatus>
        </ReportedItem>
        ... additional reported item nodes returned ...
        <ReportedItem>
            <ItemID>12345</ItemID>
            <ItemStatus>SubmissionFailed</ItemStatus>
            <ItemReasonForFailure>Item "770000368135" is invalid, not activated, or no longer in our database.</ItemReasonForFailure>
        </ReportedItem>
    </ReportedItemDetails>
</GetVeROReportStatusResponse>
```

GetVeROReportStatus Details

Prerequisites

- The requestor must be a registered eBay user.
- The requestor must be a member of the Verified Rights Owner (VeRO) Program to use this call.

GetVeROReportStatus Best Practices

Managing the Volume of the Response

VeRO report status information is maintained for two years.

Max Records Submitted/Returned

There is currently no limit on number of records returned.

Detail Levels

[GetVeROReportStatus](#) does not support multiple detail levels. It is not necessary to specify a detail level for this call.

Pagination

[GetVeROReportStatus](#) does not require pagination.

Use the [Pagination](#) input element (on the [GetVeROReportStatus](#) request) to control the pagination of data returned per call. Use [PaginationEntriesPerPage](#) to control how reported items detail entries are returned per call and [PaginationPageNumber](#) to specify which page of data to retrieve in the current call. The value specified for [PaginationEntriesPerPage](#) must be between 1 and 200.

In the output, the [PaginationTotalNumberOfEntries](#) property indicates the total number of reported items detail entries available to be returned (given the report type and date range) with multiple calls. The [PaginationTotalNumberOfPages](#) property indicates the total number of pages of data that can be returned.

Data Filters

[GetVeROReportStatus](#) provides a couple filter options reducing the volume of the response. Using [TimeFrom](#) and [TimeTo](#), you can retrieve items you reported for infringement for a specific time period (based upon the date and time the report was submitted).

Testing GetVeROReportStatus

- Testing for VeRO calls is not supported in the sandbox environment. Contact Developer Support for information on how to test in the production environment.

Related Calls

See “[GetVeROReasonCodeDetails](#)” on page 880.

See “[VeROReportItems](#)” on page 948.

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 169

GetWantItNowPost

Use [GetWantItNowPost](#) to retrieve the data for a single Want It Now post on an eBay site.

[GetWantItNowPost](#) returns the data for the post into a [WantItNowPost](#) object. The [WantItNowPost](#) object returned by this call includes all available information about the post, including the following output fields: [CategoryID](#), [Description](#), [PostID](#), [Site](#), [StartTime](#), [ResponseCount](#), and [Title](#).

When you use the [GetWantItNowSearchResults](#) call, it returns an array of [WantItNowPost](#) objects that match the search criteria. The [Description](#) data, however, is not included in [WantItNowPost](#) objects returned by the [GetWantItNowSearchResults](#) call.

- “[GetWantItNowPost Input Fields](#)” on page 889
- “[GetWantItNowPost Output Fields](#)” on page 889
- “[GetWantItNowPost Details](#)” on page 890
- “[GetWantItNowPost Best Practices](#)” on page 890
- “[Testing GetWantItNowPost](#)” on page 890
- “[GetWantItNowPost Sample](#)” on page 891
- “[Related Calls](#)” on page 892
- “[Change History](#)” on page 892

GetWantItNowPost Input Fields

See the following document for information about the call’s fields, including definitions and validation rules.



[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

GetWantItNowPost Output Fields

If the [GetWantItNowPost](#) execution is successful, it returns the data for a single Want It Now post with the specified post ID. If the call is unsuccessful, it returns an error construct (see “[Error Handling](#)” on page 107). Note that if a warning occurs, the result set will contain both normal data and an error construct.

See the following document for information about the call’s fields, including definitions and validation rules.



[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

GetWantItNowPost Details

Prerequisites

There are no special prerequisites for using this call.

Working with the Response

[GetWantItNowPost](#) returns the data for the specified Want It Now post into a [WantItNowPost](#) object. The response object includes the following fields: [CategoryID](#), [Description](#), [PostID](#), [Site](#), [StartTime](#), [ResponseCount](#), and [Title](#). Although [GetWantItNowSearchResults](#) returns most of this information, only [GetWantItNowPost](#) returns [Description](#) for a post. Review or parse the [Description](#) value for additional information to help you match your items to the post.

GetWantItNowPost Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

[GetWantItNowPost](#) returns a result set consisting of a data for a single post.

Detail Levels

[GetWantItNowPost](#) does not require or support varying detail levels.

Pagination

[GetWantItNowPost](#) does not require or support pagination.

Data Filters

[GetWantItNowPost](#) does not provide data filters for reducing the volume of the response.

Reducing the Number of Calls You Make

Rather than using [GetWantItNowPost](#) repeatedly to retrieve the data for individual posts, use [GetWantItNowSearchResults](#) to retrieve data for all posts matching your specified criteria.

Testing GetWantItNowPost

Before a Want It Now post can be retrieved from the Sandbox environment, that post must exist. If necessary, create a Want It Now post through the Sandbox web interface. There are no API calls for creating Want It Now posts.

You must specify **PostID** when you use **GetWantItNowPost**. You can get a **PostID** (post number) for a Want It Now post by using **GetWantItNowSearchResults** or by opening a post on the simulated eBay site in the Sandbox.

GetWantItNowPost Sample

The samples in this section were created using the XML API. If you are working with the SOAP API, you can still use these samples as a guide, because they are equivalent to the structure in the body of a SOAP request and response.

Please note that these samples were created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if one of these samples does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in these samples.

The following sample shows how to retrieve an item using the default detail level:

Example 169-1 GetWantItNowPost Input Sample

```
<?xml version="1.0" encoding="utf-8"?>
<GetWantItNowPostRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <RequesterCredentials>
        <eBayAuthToken>YOUR TOKEN HERE</eBayAuthToken>
    </RequesterCredentials>
    <PostID>4076143368</PostID>
</GetWantItNowPostRequest>
```

Here is the output from making that call. Additional fields may be returned in the response if they were added to the schema since this example was created (see the value of **Timestamp**).

Example 169-2 GetWantItNowPost Output Sample

```
<?xml version="1.0" encoding="utf-8"?>
<GetWantItNowPostResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2005-01-15T01:18:26.204Z</Timestamp>
    <Ack>Success</Ack>
    <CorrelationID>00000000-00000000-00000000-00000000-00000000-00000000-000000000000</CorrelationID>
    <Version>393</Version>
    <Build>20050110220901</Build>
    <WantItNowPost>
        <CategoryID>88443</CategoryID>
        <Description>Looking for late 70's Motobecane Le Champion (Team Champion). Complete bicycle with 56cm frame. Must be in excellent condition, with original parts and paint.</Description>
        <PostID>4076143368</PostID>
        <Site>US</Site>
        <StartTime>2005-10-28T20:08:39.000Z</StartTime>
        <ResponseCount>2</ResponseCount>
        <Title>56cm Motobecane Team Champion</Title>
    </WantItNowPost>
</GetWantItNowPostResponse>
```

Related Calls

[“GetWantItNowSearchResults” on page 893](#)
[“RespondToWantItNowPost” on page 913](#)

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 170

GetWantItNowSearchResults

[GetWantItNowSearchResults](#) searches for active Want It Now posts based on user-defined criteria.

Use [GetWantItNowSearchResults](#) to retrieve a list of Want It Now posts matching a set of criteria defined by the application or the application's end user. The properties of the [GetWantItNowSearchResults](#) request object represent a set of criteria used to build a query for finding Want It Now posts.

Use [GetWantItNowSearchResults](#) to search for posts using one or both of the following approaches:

- Free-text keyword queries, including wildcards (see [Searching by Keywords](#) in the Finding API User's Guide).
- Category ID-based queries (see the [FindProducts call reference](#) in the Shopping API.)

[GetWantItNowSearchResults](#) returns the data for the search in a [WantItNowPostArray](#) object. The [WantItNowPostArray](#) in the call response may contain zero, one, or multiple [WantItNowPost](#) objects – depending on how many Want It Now posts matched the specified search criteria. Each [WantItNowPost](#) object contains data for a single Want It Now post.

- ["GetWantItNowSearchResults Input Fields"](#) on page 893
- ["GetWantItNowSearchResults Output Fields"](#) on page 894
- ["GetWantItNowSearchResults Details"](#) on page 894
- ["GetWantItNowSearchResults Best Practices"](#) on page 895
- ["Testing GetWantItNowSearchResults"](#) on page 895
- ["GetWantItNowSearchResults Samples"](#) on page 896
- ["Related Calls"](#) on page 898
- ["Change History"](#) on page 899

GetWantItNowSearchResults Input Fields

See the following document for information about the call's fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see "[Schema Reference Documentation](#)" on page 967.

GetWantItNowSearchResults Output Fields

If the [GetWantItNowSearchResults](#) execution is successful, it returns an array containing the Want It Now posts matching the specified search criteria. When data pagination is specified in the request, the call returns a subset of the total number of posts that could be returned using the given search criteria.

If the call is unsuccessful, it returns an error construct (see “[Error Handling](#)” on page 107). Note that if a warning occurs, the result set will contain both normal data and an error construct.

See the following document for information about the call’s fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

GetWantItNowSearchResults Details

Prerequisites

There are no special prerequisites for using this call.

Working with the Response

[GetWantItNowSearchResults](#) returns data for the Want It Now posts matching the search criteria specified as input. The response contains a [WantItNowPostArray](#) object, which, in turn, contains a [WantItNowPost](#) object for each post found matching the specified search criteria.

Each [WantItNowPost](#) object returned with this call contains the following data: [CategoryID](#), [PostID](#), [StartTime](#), [ResponseCount](#), [Site](#), and [Title](#). To get the post description ([Description](#)), you must use [GetWantItNowPost](#) to retrieve individual posts.

If data pagination is used, the result set provides information about how many posts were found and how many posts were actually returned in the current call. The result set for a given call is referred to as a “page” of data. The [ItemsPerPage](#) object indicates the number of posts returned for the current call (i.e., the number of [WantItNowPost](#) objects contained in the [WantItNowPostArray](#)). The [PageNumber](#) indicates the number of the page of data that was returned. The [HasMoreItems](#) object indicates whether there additional pages of data to return. Within the [PaginationResult](#) object of the result set, the [TotalNumberOfEntries](#) indicates the total number of posts that match the search criteria, and the [TotalNumberOfPages](#) property indicates the total number of pages of data there are (the number of [GetWantItNowSearchResults](#) calls needed to return all found items).

An application should track the value specified for [Pagination.PageNumber](#) in each request object and increment this value in each succeeding call to sequentially retrieve each page of data and in the correct order. To retrieve all of the Want It Now posts that match the search criteria, the application should call [GetWantItNowSearchResults](#) repeatedly until either the value of [Pagination.PageNumber](#)

reaches the value in [PaginationResult.TotalNumberOfPages](#) or [HasMoreItems](#) in the result set returns a value of false. If Want It Now posts matching the search criteria are added between succeeding calls, they can be missed, so it is best to minimize the time between calls.

GetWantItNowSearchResults Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

[GetWantItNowSearchResults](#) can return a maximum of 400 item listings.

Detail Levels

[GetWantItNowSearchResults](#) does not require or support varying detail levels.

Pagination

Use [Pagination](#) in the [GetWantItNowSearchResults](#) object to control the pagination of data returned per call. Use [PaginationEntriesPerPage](#) to control how many Want It Now posts are returned per call. You can specify up to 400 posts per call. The valid range for this property is 1 to 400 and the default is 100.

Use [Pagination pageNumber](#) to specify which page of data to retrieve in the current call. The default value is 1. The highest value that should be specified for [Pagination pageNumber](#) in a call is context-dependent and varies from one search scenario to the next. It should not exceed the value returned in the [PaginationResult.TotalNumberOfPages](#) object of the [GetWantItNowSearchResults](#) response object.

In the output, the [PaginationResult.TotalNumberOfEntries](#) indicates how many entries could be returned. [HasMoreItems](#), if true, indicates that there are more items to be returned, requiring one or more additional calls. [PageNumber](#) indicates the page of data returned and [ItemsPerPage](#) indicates the number of items returned in the current call. The values for [PageNumber](#) and [ItemsPerPage](#) should be the same as the values specified for the corresponding input objects, except in the case of the last page, where there can be fewer than the maximum-per-call number.

Data Filters

[GetWantItNowSearchResults](#) does not provide data filters, such as date ranges, but you can limit the number of posts you get in your result set by refining your query and/or specifying a category to search. See [Searching by Keywords](#) in the Finding API User's Guide and the [FindProducts call reference](#) in the Shopping API.

Testing GetWantItNowSearchResults

To test [GetWantItNowSearchResults](#) in the Sandbox environment, you should have access to multiple Want It Now posts with titles that the developer knows will be returned with a given search query

string. If such posts do not already exist, you must create them through the Sandbox web interface. There are no API calls for creating Want It Now posts.

- Be sure to set up at least one buyer and at least one seller in the Sandbox environment.
- As a buyer, create Want It Now posts on the simulated eBay site in the Sandbox. Use specific keywords to make your posts easily identifiable. Add posts to different categories.
- As a seller, use [GetWantItNowSearchResults](#) to search for your posts.

GetWantItNowSearchResults Samples

The samples in this section were created using the XML API. If you are working with the SOAP API, you can still use these samples as a guide, because they are equivalent to the structure in the body of a SOAP request and response.

Please note that these samples were created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if one of these samples does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in these samples.

["Search by Category"](#) on page 896
["Keyword Query"](#) on page 897

Search by Category

The following sample shows querying for posts in a category. Pagination is used to handle a large result set. This sample queries for posts in the books category (category ID 267):

Example 170-1 GetWantItNowSearchResults Sample Input

```
<?xml version="1.0" encoding="utf-8"?>
<GetWantItNowSearchResultsRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <RequesterCredentials>
        <eBayAuthToken>YOUR TOKEN HERE</eBayAuthToken>
    </RequesterCredentials>
    <CategoryID>267</CategoryID>
</GetWantItNowSearchResultsRequest>
```

This is the output from the call:

Example 170-2 GetWantItNowSearchResults Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<GetWantItNowSearchResultsResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2005-01-27T00:58:56.300Z</Timestamp>
    <Ack>Success</Ack>
    <Version>433</Version>
    <Build>20050123212033</Build>
    <WantItNowPostArray>
```

Example 170-2 GetWantItNowSearchResults Sample Output

```

<WantItNowPost>
  <CategoryID>279</CategoryID>
  <Site>US</Site>
  <PostID>4100522667</PostID>
  <StartTime>2005-10-24T23:14:17.000Z</StartTime>
  <ResponseCount>3</ResponseCount>
  <Title>Tales of Uncle Remus</Title>
</WantItNowPost>
<WantItNowPost>
  <CategoryID>29792</CategoryID>
  <Site>US</Site>
  <PostID>4100543214</PostID>
  <StartTime>2005-10-24T23:14:17.000Z</StartTime>
  <ResponseCount>0</ResponseCount>
  <Title>Homer's Daughter Audio Book by Robert Graves</Title>
</WantItNowPost>
<WantItNowPost>
  <CategoryID>280</CategoryID>
  <Site>Canada</Site>
  <PostID>4100551573</PostID>
  <StartTime>2005-10-24T23:14:17.000Z</StartTime>
  <ResponseCount>0</ResponseCount>
  <Title>Hooked On Crochet March/April 1987 Issue #2</Title>
</WantItNowPost>
  ... additional Want It Now post nodes returned ...
</WantItNowPostArray>
<HasMoreItems>true</HasMoreItems>
<ItemsPerPage>50</ItemsPerPage>
<PageNumber>1</PageNumber>
<PaginationResult>
  <TotalNumberOfPages>114</TotalNumberOfPages>
  <TotalNumberOfEntries>5689</TotalNumberOfEntries>
</PaginationResult>
</GetWantItNowSearchResultsResponse>

```

Keyword Query

The following sample shows querying for “easy bake oven” posts on all eBay sites:

Example 170-3 GetWantItNowSearchResults Sample Input

```

<?xml version="1.0" encoding="utf-8"?>
<GetWantItNowSearchResultsRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>YOUR TOKEN HERE</eBayAuthToken>
  </RequesterCredentials>
  <Query>easy bake oven</Query>
  <SearchWorldwide>True</SearchWorldwide>
</GetWantItNowSearchResultsRequest>

```

This is the output from the call:

Example 170-4 GetWantItNowSearchResults Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<GetWantItNowSearchResultsResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-01-27T00:58:56.300Z</Timestamp>
  <Ack>Success</Ack>
  <Version>433</Version>
  <Build>20050123212033</Build>
  <WantItNowPostArray>
    <WantItNowPost>
      <CategoryID>19169</CategoryID>
      <Site>US</Site>
      <PostID>4100551160</PostID>
      <StartTime>2005-10-24T23:14:17.000Z</StartTime>
      <ResponseCount>0</ResponseCount>
      <Title>Easy-Bake oven accessories</Title>
    </WantItNowPost>
    <WantItNowPost>
      <CategoryID>19169</CategoryID>
      <Site>US</Site>
      <PostID>4100551158</PostID>
      <StartTime>2005-10-24T23:14:17.000Z</StartTime>
      <ResponseCount>0</ResponseCount>
      <Title>vintage betty crocker cake mixes for easy bake oven</Title>
    </WantItNowPost>
    <WantItNowPost>
      <CategoryID>19169</CategoryID>
      <Site>US</Site>
      <PostID>4100551156</PostID>
      <StartTime>2005-10-24T23:14:17.000Z</StartTime>
      <ResponseCount>1</ResponseCount>
      <Title>Easy Bake accessories, pans</Title>
    </WantItNowPost>
    <WantItNowPost>
      <CategoryID>19169</CategoryID>
      <Site>UK</Site>
      <PostID>4300023432</PostID>
      <StartTime>2005-10-24T23:14:17.000Z</StartTime>
      <ResponseCount>0</ResponseCount>
      <Title>Easy Bake Oven Pans</Title>
    </WantItNowPost>
  </WantItNowPostArray>
</GetWantItNowSearchResultsResponse>
```

Related Calls

- “[GetWantItNowPost](#)” on page 889
- “[RespondToWantItNowPost](#)” on page 913
- “[GetSearchResults](#)” on page 856

Change History

For information about changes that have been made to this call, see the call's "Change History" section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 171

IssueRefund

Use **IssueRefund** to issue a refund to a buyer for a single Half.com transaction. Refunds may only be issued for a specific transaction. Sellers do not have the ability to issue a general refund (not tied to a transaction) to a buyer.

To issue multiple refunds, execute **IssueRefund** once for each refund. **IssueRefund** returns the amount the seller sent to Half.com to refund to the buyer, and the total amount that Half.com refunded to the buyer.

For basic information about Half.com listings and payments, see “[Half.com](#)” on page 574.

Important: The Sandbox does not support Half.com.

- “[IssueRefund Input Fields](#)” on page 900
- “[IssueRefund Output Fields](#)” on page 900
- “[IssueRefund Details](#)” on page 901
- “[IssueRefund Best Practices](#)” on page 901
- “[IssueRefund Samples](#)” on page 902
- “[Related Calls](#)” on page 902
- “[Change History](#)” on page 902

IssueRefund Input Fields

IssueRefund takes an item ID, transaction ID, and refund amount as input, among other fields.

See the following document for information about the call’s fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

IssueRefund Output Fields

If the execution of **IssueRefund** is successful, it retrieves information about pending or paid payments from Half.com to the seller.

See the following document for information about the call's fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

If unsuccessful, the call returns an error construct (see “[Error Handling](#)” on page 107). Note that if a warning occurs, the result set will contain both normal data and an error construct.

IssueRefund Details

Prerequisites

See “[Working with Half.com Listings](#)” on page 558 for requirements that sellers need to meet in order issue refunds using **IssueRefund**.

Working with the Response

IssueRefund retrieves the amount that was refunded to the buyer.

IssueRefund Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

IssueRefund only returns a result set for one refund.

Detail Levels

IssueRefund does not support multiple detail levels. It is not necessary to specify a detail level for this call.

Pagination

IssueRefund does not require pagination.

Data Filters

IssueRefund does not provide data filters for reducing the volume of the response.

Reducing the Number of Calls You Make

There are no special features to reduce the number of times you execute **IssueRefund**.

IssueRefund Samples

The samples in this section were created using the XML API. If you are working with the SOAP API, you can still use these samples as a guide, because they are equivalent to the structure in the body of a SOAP request and response.

Please note that these samples were created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if one of these samples does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in these samples.

The following sample shows a request that issues a refund. This request was submitted to site ID 0 (the US site).

Example 171-1 IssueRefund Sample Input

```
<?xml version="1.0" encoding="utf-8"?>
<IssueRefundRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <Version>431</Version>
  <RequesterCredentials>
    <eBayAuthToken>YOURTOKENHERE</eBayAuthToken>
  </RequesterCredentials>
  <ItemID>14014825444</ItemID>
  <TransactionID>29337</TransactionID>
  <RefundMessage>Sorry. This item is already sold.</RefundMessage>
  <RefundReason>CannotShipProduct</RefundReason>
  <RefundType>Full</RefundType>
</IssueRefundRequest>
```

Related Calls

[“GetSellerPayments” on page 860](#)

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 172

LeaveFeedback

Use this call to leave feedback about another user at the conclusion of a successful sales transaction (item sold).

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[LeaveFeedback](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 173

PlaceOffer

Use this call to make a bid on an item, a best offer, or an item purchase.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call, including information on PlaceOffer Policies and Guidelines:

[PlaceOffer](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 174

RelistFixedPriceItem

Use this call to relist a fixed-price item on eBay.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[RelistFixedPriceItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 175

RelistItem

Relists to the eBay site a listing that previously ended.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[RelistItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 176

RemoveFromWatchList

Use this call to remove one or more items from the user's My eBay watch list.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[RemoveFromWatchList](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 177

RespondToBestOffer

[RespondToBestOffer](#) enables the seller of a Best Offer item to respond with Accept, Decline, or Counter to offers made by bidders. See “[Enabling Best Offer](#)” on page 225.

- “[RespondToBestOffer Input Fields](#)” on page 908
- “[RespondToBestOffer Output Fields](#)” on page 908
- “[RespondToBestOffer Samples](#)” on page 908
- “[RespondToBestOffer Details](#)” on page 910
- “[RespondToBestOffer Best Practices](#)” on page 910
- “[Testing RespondToBestOffer](#)” on page 911
- “[Related Calls](#)” on page 911
- “[Change History](#)” on page 911

RespondToBestOffer Input Fields

See the following document for information about the call’s fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

RespondToBestOffer Output Fields

See the following document for information about the call’s fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

RespondToBestOffer Samples

The samples in this section were created using the XML API. If you are working with the SOAP API, you can still use these samples as a guide, because they are equivalent to the structure in the body of a SOAP request and response.

Please note that these samples were created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if one of these samples does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in these samples.

The following request declines all best offers made for a particular item.

Example 177-1 RespondToBestOffer Sample Input

```
<?xml version="1.0" encoding="utf-8"?>
<RespondToBestOfferRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <SiteId>0</SiteId>
  <ItemID>1509248245</ItemID>
  <BestOfferID>21</BestOfferID>
  <BestOfferID>28</BestOfferID>
  <Action>Decline</Action>
  <SellerResponse>Your offer is, like, way too low, dude</SellerResponse>
</RespondToBestOfferRequest>
```

Here is the output from making that call:

Example 177-2 RespondToBestOffer Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<RespondToBestOfferResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-02-02T23:56:18.840Z</Timestamp>
  <Ack>Success</Ack>
  <CorrelationID>00000000-00000000-00000000-
    00000000-00000000-00000000-0000000000</CorrelationID>
  <Version>399</Version>
  <Build>20050131190307</Build>
  <RespondToBestOffer>
    <BestOffer>
      <Id>21</Id>
      <Status>Success</Status>
    </BestOffer>
    <BestOffer>
      <Id>28</Id>
      <Status>Success</Status>
    </BestOffer>
  </RespondToBestOffer>
</RespondToBestOfferResponse>
```

The following request makes a counteroffer for a best offers made for a particular item.

Example 177-3 RespondToBestOffer Sample Input

```
<?xml version="1.0" encoding="utf-8"?>
<RespondToBestOfferRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <RequesterCredentials>
        <eBayAuthToken>ABC...123</eBayAuthToken>
    </RequesterCredentials>
    <SiteId>0</SiteId>
    <ItemID>2214021994</ItemID>
    <BestOfferID>37647</BestOfferID>
    <Action>Counter</Action>
    <CounterOfferPrice>7</CounterOfferPrice>
    <CounterOfferQuantity>1</CounterOfferQuantity>
    <SellerResponse>Your offer is too low, here's the lowest I'll accept</SellerResponse>
</RespondToBestOfferRequest>
```

Here is the output from making that call:

Example 177-4 RespondToBestOffer Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<RespondToBestOfferResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2005-12-15T23:41:25.419Z</Timestamp>
    <Ack>Success</Ack>
    <Version>439</Version>
    <Build>20050131190307</Build>
    <RespondToBestOffer>
        <BestOffer>
            <Status>Success</Status>
        </BestOffer>
    </RespondToBestOffer>
</RespondToBestOfferResponse>
```

RespondToBestOffer Details

See “[Enabling Best Offer](#)” on page 225.

RespondToBestOffer Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

As many best offer statuses are returned as there were best offers rejected or accepted in the request.

Detail Levels

RespondToBestOffer has no additional detail levels.

Pagination

RespondToBestOffer does not require pagination.

Data Filters

RespondToBestOffer does not provide data filters for reducing the volume of the response.

Reducing the Number of Calls You Make

Best offers can be declined in bulk, using the same message from the seller to the bidders of all rejected offers.

Testing RespondToBestOffer

- As of the time of this release, testing is not currently supported in the sandbox environment. Contact Developer Support for information on how to test in the production environment.

Related Calls

See “[GetBestOffers](#)” on page 803.

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 178

RespondToFeedback

Use this call to reply to feedback that has been left for a user, or to post a follow-up comment to a feedback comment a user has left for someone else.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[RespondToFeedback](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 179

RespondToWantItNowPost

Use **RespondToWantItNowPost** to respond to a Want It Now post with an item listed on the eBay site. Responses appear on the Want It Now post page, with the item title, the price of the item, the number of bids on the item, and the time left before the listing ends. If the item has a picture, the picture is also included on the Want It Now post page.

- “[RespondToWantItNowPost Input Fields](#)” on page 913
- “[RespondToWantItNowPost Output Fields](#)” on page 913
- “[RespondToWantItNowPost Details](#)” on page 914
- “[RespondToWantItNowPost Best Practices](#)” on page 914
- “[Testing RespondToWantItNowPost](#)” on page 914
- “[RespondToWantItNowPost Sample](#)” on page 914
- “[Related Calls](#)” on page 915
- “[Change History](#)” on page 915

RespondToWantItNowPost Input Fields

See the following document for information about the call’s fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

RespondToWantItNowPost Output Fields

If the **RespondToWantItNowPost** execution is successful, it returns the fields of the abstract response object (there are no call-specific return fields). If unsuccessful, it returns an error construct (see “[Error Handling](#)” on page 107). Note that if a warning occurs, the result set will contain both normal data and an error construct.

See the following document for information about the call’s fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

RespondToWantItNowPost Details

Prerequisites

There are no special prerequisites for using this call.

Working with the Response

There are no call-specific fields in the response.

RespondToWantItNowPost Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

[RespondToWantItNowPost](#) does not return any call-specific data.

Detail Levels

[RespondToWantItNowPost](#) does not require or support varying detail levels.

Pagination

[RespondToWantItNowPost](#) does not require pagination.

Data Filters

[RespondToWantItNowPost](#) does not provide data filters for reducing the volume of the response.

Reducing the Number of Calls You Make

There is no alternate API call that can be used to respond to Want It Now posts.

Testing RespondToWantItNowPost

To test [RespondToWantItNowPost](#) in the Sandbox, you must have the post ID for the Want It Now post you are responding to, and the item ID for the item listing you want to provide as a response.

The call will result in an error if either the post ID or the item ID are not valid. An error also occurs if you attempt to respond to a Want It Now post with an item ID that has already been submitted as a response to the specified post.

RespondToWantItNowPost Sample

The sample in this section was created using the XML API. If you are working with the SOAP API, you can still use this sample as a guide, because it is equivalent to the structure in the body of a SOAP request and response.

Please note that this sample was created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if this sample does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in the sample.

The following sample shows the input for the call to respond to a Want It Now post:

Example 179-1 RespondToWantItNowPost Sample Input

```
<RespondToWantItNowPostRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>YOUR TOKEN HERE</eBayAuthToken>
  </RequesterCredentials>
  <PostID>4076134467</PostID>
  <ItemID>9200829272</ItemID>
</RespondToWantItNowPostRequest>
```

Here is the output from making that call:

Example 179-2 RespondToWantItNowPost Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<RespondToWantItNowPostResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-05-25T16:54:01.711Z</Timestamp>
  <Ack>Success</Ack>
  <Message>Your response may take up to several hours to appear.</Message>
  <Version>433</Version>
  <Build>20050524220521</Build>
</RespondToWantItNowPostResponse>
```

Related Calls

- “[GetWantItNowSearchResults](#)” on page 893
- “[GetWantItNowPost](#)” on page 889
- “[GetSearchResults](#)” on page 856

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 180

ReviseCheckoutStatus

Use this call to update the payment details and status of a transaction or order.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[ReviseCheckoutStatus](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 181

ReviseFixedPriceItem

Use this call to revise an active fixed-price item listing on eBay.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[ReviseFixedPriceItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 182

ReviseItem

This call allows a seller to make changes to a currently active item listing.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[ReviseItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 183

ReviseMyMessages

Use this call to change the read and flagged states for a given message.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[ReviseMyMessages](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 184

ReviseMyMessagesFolders

Use this call to show, rename, or delete specified My Messages folders for a given user. The Inbox and Sent folders cannot be revised.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[ReviseMyMessagesFolders](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 185

RevokeToken

Use this call to revoke a token before it is scheduled to expire.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[RevokeToken](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 186

SellerReverseDispute

Allows a seller to reverse an Unpaid Item dispute that has already been closed.

- “[Overview](#)” on page 922
- “[SellerReverseDispute Input Fields](#)” on page 922
- “[SellerReverseDisputeCall Output Fields](#)” on page 922
- “[SellerReverseDispute Best Practices](#)” on page 923
- “[Testing SellerReverseDispute](#)” on page 923
- “[SellerReverseDispute Samples](#)” on page 923
- “[Related Calls](#)” on page 925
- “[Change History](#)” on page 925

Overview

SellerReverseDispute enables a seller to "reverse" an Unpaid Item dispute that has been closed, for example, if buyer and seller reach an agreement. The seller's Final Value Fee credit and the buyer's strike are both reversed. The dispute might have resulted in a strike to the buyer and a Final Value Fee credit to the seller. A buyer and seller sometimes come to agreement after a dispute has been closed. In particular, the seller might discover that the buyer actually paid, or the buyer might agree to pay the seller's fees in exchange for having the strike removed.

See “[Unpaid Item Disputes](#)” on page 455. (This call is not used for Item Not Received disputes.)

SellerReverseDispute Input Fields

See the following document for information about the call's fields, including definitions and validation rules.

[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

SellerReverseDisputeCall Output Fields

If the call to **SellerReverseDispute** is successful, it returns a call status of **Success**. If unsuccessful, it returns an error construct (see “[Error Handling](#)” on page 107).

Note that if a warning occurs, the result will contain both a call status and an error.

SellerReverseDispute Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

[SellerReverseDispute](#) returns a call status or an error. It does not return data records.

Detail Levels

[SellerReverseDispute](#) does not support multiple detail levels. It is not necessary to specify a detail level for this call.

Pagination

[SellerReverseDispute](#) does not require pagination.

Data Filters

[SellerReverseDispute](#) does not provide data filters for reducing the volume of the response.

Reducing the Number of Calls You Make

It is not necessary to reduce the number of calls you make to [SellerReverseDispute](#).

Testing SellerReverseDispute

- Set up at least one buyer and at least one seller in the Sandbox.
- See “[Testing Unpaid Item Disputes in the Sandbox](#)” on page 471 for an illustration of creating a dispute. The kind of dispute needed for testing [SellerReverseDispute](#) must be closed with a [DisputeActivity](#) of [SellerEndCommunication](#), [CameToAgreementNeedFVFCredit](#), or [MutualAgreementOrNoBuyerResponse](#).
- Call [GetDispute](#) and ensure that the [DisputeState](#) is [Closed](#).
- Call [SellerReverseDispute](#) and ensure [CallStatus](#) is [Success](#).
- Call [GetDispute](#) and check the value of [DisputeState](#) to confirm that it was reversed.

SellerReverseDispute Samples

The samples in this section were created using the XML API. If you are working with the SOAP API, you can still use these samples as a guide, because they are equivalent to the structure in the body of a SOAP request and response.

Please note that these samples were created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if one of these samples does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in these samples.

[Example 186-1](#) shows a request to [SellerReverseDispute](#) to reverse an Unpaid Item dispute closed with a [DisputeActivity](#) of [SellerEndCommunication](#).

Example 186-1 SellerReverseDispute Sample Request

```
<?xml version="1.0" encoding="utf-8"?>
<SellerReverseDisputeRequest xmlns="urn:ebay:apis:eBLBaseComponents">
  <RequesterCredentials>
    <eBayAuthToken>ABC...123</eBayAuthToken>
  </RequesterCredentials>
  <DisputeID>213833</DisputeID>
  <DisputeResolutionReason>OtherResolution</DisputeResolutionReason>
</SellerReverseDisputeRequest>
```

[Example 186-2](#) is the response from [SellerReverseDispute](#) showing that the dispute was successfully reversed.

Example 186-2 SellerReverseDispute Sample Response

```
<?xml version="1.0" encoding="utf-8"?>
<SellerReverseDisputeResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-03-17T01:48:54.152Z</Timestamp>
  <Ack>Success</Ack>
  <CorrelationID>
    00000000-00000000-00000000-00000000-00000000-00000000-0000000000
  </CorrelationID>
  <Version>401</Version>
  <Build>20050308050919</Build>
</SellerReverseDisputeResponse>
```

After calling [SellerReverseDispute](#), this call to [GetDispute](#) ([Example 186-3](#)) shows a [DisputeStatus](#) of [StrikeAppealedAndFVFCreditReversed](#). You can also check the [DisputeResolution](#) elements to see that the buyer's strike was appealed and the seller's credit was reversed.

Example 186-3 Response from GetDispute After the Dispute Was Reversed

```
<?xml version="1.0" encoding="utf-8"?>
<GetDisputeResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2005-03-17T02:04:45.309Z</Timestamp>
  <Ack>Success</Ack>
  <CorrelationID>
    00000000-00000000-00000000-00000000-00000000-00000000-0000000000
  </CorrelationID>
  <Version>401</Version>
  <Build>20050308050919</Build>

  <Dispute>
    <DisputeID>213833</DisputeID>
    <DisputeRecordType>UnpaidItem</DisputeRecordType>
    <DisputeState>Closed</DisputeState>
    <DisputeStatus>StrikeAppealedAndFVFCreditReversed</DisputeStatus>
    <BuyerUserID>samplebuyer</BuyerUserID>
    <SellerUserID>sampleseller</SellerUserID>
    <TransactionID>748970</TransactionID>

    <Item>
      <ItemID>9202501086</ItemID>
      <ListingDetails>
        <StartTime>2005-02-08T02:37:07.000Z</StartTime>
        <EndTime>2005-02-08T02:39:43.000Z</EndTime>
      </ListingDetails>
      <Quantity>1</Quantity>
      <SellingStatus>
        <ConvertedCurrentPrice currencyID="USD">199.0</ConvertedCurrentPrice>
        <CurrentPrice currencyID="USD">199.0</CurrentPrice>
      </SellingStatus>
      <Site>US</Site>
      <Title>Antique Silver Candelabra</Title>
    </Item>
```

Example 186-3 Response from GetDispute After the Dispute Was Reversed

```
<DisputeReason>BuyerHasNotPaid</DisputeReason>
<DisputeExplanation>BuyerHasNotResponded</DisputeExplanation>
<DisputeCreditEligibility>Ineligible</DisputeCreditEligibility>
<DisputeCreatedTime>2005-02-09T23:50:38.000Z</DisputeCreatedTime>
<DisputeModifiedTime>2005-03-17T01:48:54.000Z</DisputeModifiedTime>

<DisputeResolution>
  <DisputeResolutionRecordType>StrikeBuyer</DisputeResolutionRecordType>
  <ResolutionTime>2005-03-17T01:41:44.000Z</ResolutionTime>
</DisputeResolution>

<DisputeResolution>
  <DisputeResolutionRecordType>FVFCredit</DisputeResolutionRecordType>
  <ResolutionTime>2005-03-17T01:41:43.000Z</ResolutionTime>
</DisputeResolution>

<DisputeResolution>
  <DisputeResolutionRecordType>AppealBuyerStrike</DisputeResolutionRecordType>
  <DisputeResolutionReason>OtherResolution</DisputeResolutionReason>
  <ResolutionTime>2005-03-17T01:48:53.000Z</ResolutionTime>
</DisputeResolution>

<DisputeResolution>
  <DisputeResolutionRecordType>ReverseFVFCredit</DisputeResolutionRecordType>
  <DisputeResolutionReason>OtherResolution</DisputeResolutionReason>
  <ResolutionTime>2005-03-17T01:48:54.000Z</ResolutionTime>
</DisputeResolution>

<DisputeMessage>
  <MessageID>431164</MessageID>
  <MessageSource>Seller</MessageSource>
  <MessageCreationTime>2005-03-17T01:40:59.000Z</MessageCreationTime>
  <MessageText>Last attempt to contact the buyer</MessageText>
</DisputeMessage>

<DisputeMessage>
  <MessageID>431165</MessageID>
  <MessageSource>eBay</MessageSource>
  <MessageCreationTime>2005-03-17T01:41:46.000Z</MessageCreationTime>
  <MessageText>
    The Unpaid Item dispute has been closed for the following reason:
    The seller has ended this communication and filed for a Final Value Fee credit.
  </MessageText>
</DisputeMessage>

<Escalation>false</Escalation>
<PurchaseProtection>false</PurchaseProtection>
</Dispute>
</GetDisputeResponse>
```

Related Calls

- “AddDispute” on page 771
- “AddDisputeResponse” on page 775
- “GetDispute” on page 828
- “ GetUserDisputes” on page 878

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Chapter 187

SendInvoice

Use this call to send an email invoice or order to a buyer, or order information to a seller.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SendInvoice](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 188

SetMessagePreferences

Use this call to add custom Ask Seller a Question (ASQ) subjects to display on the seller's Ask a Question page, or reset custom subjects to their default values.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SetMessagePreferences](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 189

SetNotificationPreferences

This call is part of eBay Platform Notifications, an optional feature that alerts applications and/or users when certain events occur on the eBay site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SetNotificationPreferences](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 190

SetPromotionalSale

Use this call to create or modify a promotional sale.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[setPromotionalSale](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 191

SetPromotionalSaleListings

Use this call to change which item listings are affected by a promotional sale.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SetPromotionalSaleListings](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 192

SetPromotionRules

SetPromotionRules has been deprecated. It is no longer functional as of June 2007.

Chapter 193

SetShippingDiscountProfiles

Use this call to define shipping discount profiles and to define combined payment discounts for packaging and handling costs.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SetShippingDiscountProfiles](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 194

SetStore

Use this call to set the configuration of a user's eBay store.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SetStore](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 195

SetStoreCategories

Use this call to modify the category structure of an eBay Store.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SetStoreCategories](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 196

SetStoreCustomPage

Use this call to set a custom page for a user's eBay Store.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SetStoreCustomPage](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 197

SetStorePreferences

Use this call to set the preferences for a user's eBay Store.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SetStorePreferences](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 198

SetTaxTable

This call sets the tax table for the specified seller for the specified site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SetTaxTable](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 199

SetUserNotes

Use this call to add notes to items that are being tracked in a user's My eBay page, including items the user is selling, items the user is bidding on, and items in the user's watch list.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SetUserNotes](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 200

SetUserPreferences

Use this call to set the user preferences for a single user registered on an eBay site.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[SetUserPreferences](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 201

UploadSiteHostedPictures

Use this call to upload a picture to eBay Picture Services (EPS) by including a binary attachment or by pointing to an image on a web server.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[UploadSiteHostedPictures](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 202

ValidateChallengeInput

Use this call to validate the user response to a botblock challenge.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[ValidateChallengeInput](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 203

ValidateTestUserRegistration

Use this call to provide settings for a test user and to enable a manually-registered test user to list items in the Sandbox environment.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[ValidateTestUserRegistration](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 204

VerifyAddFixedPriceItem

Use this call to test the insertion of a fixed-price item on eBay without actually listing the item.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[VerifyAddFixedPriceItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 205

VerifyAddItem

Use this call to test the definition of a new item without listing the item.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[VerifyAddItem](#)

[Input](#)

[Output](#)

[Samples](#)

Sellers who engage in cross-border trade on sites that require a recoupment agreement, must agree to the recoupment terms before adding or verifying items on the site. This agreement allows eBay to reimburse a buyer during a dispute and then recoup the cost from the seller.

Chapter 206

VerifyAddSecondChanceItem

Tests the inputs for and [AddSecondChanceItem](#) call without actually extending the second chance offer.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[VerifyAddSecondChanceItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 207

VerifyRelistItem

Use this call to check arguments for a call to [RelistItem](#) for errors and lists fees—*without actually relisting the item*.

Note: The information for this call has been consolidated into the eBay Trading API Call Reference. This section will eventually be removed from this guide.

See the following sections in the Call Reference for information about this call:

[VerifyRelistItem](#)

[Input](#)

[Output](#)

[Samples](#)

Chapter 208

VeROReportItems

Use **VeROReportItems** to report items that are infringing your copyright, trademark, or other intellectual property rights. **VeROReportItems** allows you to report one or more items at a time. You must be a member of the Verified Rights Owner (VeRO) Program to use this call.

When a reported item has been successfully processed, eBay sends a Notice of Claimed Infringement (NOCI) email to the seller of the alleged infringing item and ends the listing.

For more information about the VeRO Program and how eBay protects intellectual property rights, refer to the following help pages on the eBay site:

<http://pages.ebay.in/help/confidence/programs-vero.html>

<http://pages.ebay.com/help/tp/vero-rights-owner.html>

[“VeROReportItems Input Fields” on page 948](#)
[“VeROReportItems Output Fields” on page 948](#)
[“VeROReportItems Details” on page 949](#)
[“VeROReportItems Best Practices” on page 949](#)
[“Testing VeROReportItems” on page 949](#)
[“VeROReportItems Sample” on page 949](#)
[“Related Calls” on page 951](#)
[“Change History” on page 951](#)

VeROReportItems Input Fields

See the following document for information about the call’s fields, including definitions and validation rules.



[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see [“Schema Reference Documentation” on page 967](#).

VeROReportItems Output Fields

See the following document for information about the call’s fields, including definitions and validation rules.



[eBay Trading API Call Reference](#)

If you are looking for an earlier style of the WSDL or Schema documentation, or the Visual Schema documentation, please see “[Schema Reference Documentation](#)” on page 967.

VeROReportItems Details

Prerequisites

- The requestor must be a registered eBay user.
- The requestor must be a member of the Verified Rights Owner (VeRO) Program to use this call.

VeROReportItems Best Practices

Managing the Volume of the Response

Max Records Submitted/Returned

There is currently no limit on number of records returned.

Detail Levels

VeROReportItems does not support multiple detail levels. It is not necessary to specify a detail level for this call.

Pagination

VeROReportItems does not require pagination.

Data Filters

VeROReportItems does not provide data filters for reducing the volume of the response.

Testing VeROReportItems

- Testing for VeRO calls is not supported in the sandbox environment. Contact Developer Support for information on how to test in the production environment.

VeROReportItems Sample

The sample in this section was created using the XML API. If you are working with the SOAP API, you can still use this sample as a guide, because it is equivalent to the structure in the body of a SOAP request and response.

Please note that this sample was created when specific items, categories, and related data were listed on the Production site. Since eBay catalog data changes often, the same data might not be active when you run these samples. Therefore, if this sample does not succeed, you may need to execute a call to find current data (listing IDs, Watch List IDs, etc.), and then substitute more recent data for the data shown in the sample.

The following call sample reports a couple item listings for infringement. One of the items is reported twice with separate reason codes. For each reported item, the VeRO Program member will be copied on the NOCI email sent to the seller.

Example 208-1 VeROReportItems Sample Input

```
<?xml version="1.0" encoding="utf-8"?>
<VeROReportItemsRequest xmlns="urn:ebay:apis:eBLBaseComponents">
    <RequesterCredentials>
        <eBayAuthToken>YOUR TOKEN HERE</eBayAuthToken>
    </RequesterCredentials>
    <Version>491</Version>
    <RightsOwnerID>VeRO USER ID</RightsOwnerID>
    <ReportItems>
        <ReportItem>
            <ItemID>220000368135</ItemID>
            <VeROReasonCodeID>9034</VeROReasonCodeID>
            <MessageToSeller>Your listing is in violation of copyright, trademark, or other
intellectual property rights.</MessageToSeller>
            <CopyEmailToRightsOwner>true</CopyEmailToRightsOwner>
        </ReportItem>
        <ReportItem>
            <ItemID>220000368135</ItemID>
            <VeROReasonCodeID>9035</VeROReasonCodeID>
            <MessageToSeller>Your listing is in violation of copyright, trademark, or other
intellectual property rights.</MessageToSeller>
            <CopyEmailToRightsOwner>true</CopyEmailToRightsOwner>
        </ReportItem>
        <ReportItem>
            <ItemID>220000368136</ItemID>
            <VeROReasonCodeID>9034</VeROReasonCodeID>
            <MessageToSeller>Your listing is in violation of copyright, trademark, or other
intellectual property rights.</MessageToSeller>
            <CopyEmailToRightsOwner>true</CopyEmailToRightsOwner>
        </ReportItem>
    </ReportItems>
</VeROReportItemsRequest>
```

Here is the output from making that call:

Example 208-2 VeROReportItems Sample Output

```
<?xml version="1.0" encoding="utf-8"?>
<VeROReportItemsResponse xmlns="urn:ebay:apis:eBLBaseComponents">
    <Timestamp>2006-12-02T23:56:18.840Z</Timestamp>
    <Ack>Success</Ack>
    <Version>491</Version>
    <Build>e491_core_Bundled_3964543_R1</Build>
    <VeROReportPacketID>5000051397</VeROReportPacketID>
    <VeROReportPacketStatus>Received</VeROReportPacketStatus>
</VeROReportItemsResponse>
```

Related Calls

See “[GetSearchResults](#)” on page 856.

See “[GetVeROReasonCodeDetails](#)” on page 880.

See “[GetVeROReportStatus](#)” on page 884.

Change History

For information about changes that have been made to this call, see the call’s “Change History” section in the *eBay Trading API Call Reference*:

[Change History](#)

Part 2

Appendices

Useful Third-Party Resources	953
Data Types	959
Compatible Application Check for Specific Calls	963
Current Compatibility Levels	965
Error Message Strings	966

Appendix A

Useful Third-Party Resources

The resources below provide information about Web services, as well as other useful topics. The definitions within some sections are taken from several of these resources.

- “[Web Services](#)” on page 953
- “[Base64 Decoding](#)” on page 956
- “[HTML Forms](#)” on page 957
- “[Internet Explorer WebBrowser Control](#)” on page 957
- “[Regular Expressions](#)” on page 957
- “[Other Resources](#)” on page 958

Web Services

Web services is a technology that allows applications to communicate with each other in a platform- and programming language-independent manner. A *Web service* is a software interface that describes a collection of operations that can be accessed over the network through standardized XML messaging. It uses protocols based on the XML language to describe an operation to execute or data to exchange with another Web service. Web services promise to increase interoperability, and lower the costs of software integration and data-sharing with partners. As they are based on simple and non-proprietary standards, Web services are designed to make it possible for computer programs to communicate directly with one another and exchange data regardless of location, operating systems, or languages. (IBM Corporation (n.d). New to Web Services. *IBM developerWorks : Web services*. IBM Corporation. Retrieved March 3, 2004, from <http://www-106.ibm.com/developerworks/webservices/newto/#1>)

Web Services Projects and References

IBM developerWorks Web Services
IBM Corporation
<http://www-136.ibm.com/developerworks/webservices>

O'Reilly Web Services
O'Reilly & Associates, Inc.
<http://webservices.oreilly.com/>

Microsoft Web Services
Microsoft Corporation
<http://msdn.microsoft.com/webservices/>

XML and Web Services
Microsoft Corporation
<http://msdn.microsoft.com/library/en-us/dnanchor/html/WebServicesAnchor.asp?frame=true>

Java Technology and Web Services
Sun Microsystems, Inc.
<http://java.sun.com/webservices/>

Apache Web Services Project
The Apache Software Foundation
<http://ws.apache.org/>

Web Services Specifications and Standards

Web Services Activity

W3C

<http://www.w3.org/2002/ws/>

OASIS Home Page

Organization for the Advancement of Structured Information Standards (OASIS)

<http://www.oasis-open.org/home/index.php>

XML.Org Home Page

OASIS

<http://www.xml.org/>

UBL

OASIS

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=UBL

ebXML

OASIS

<http://www.ebxml.org/>

Messaging Specifications Index Page

Microsoft Corporation

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/wsmsgspecindex.asp>

Web Services Journals

O'Reilly Web Services Topics
O'Reilly & Associates, Inc.
<http://webservices.xml.com/>

XML & Web Services Magazine
Fawcette Technical Publications
<http://www.fawcette.com/xmlmag/>

Web Services Architect
<http://www.webservicesarchitect.com>

Dr. Dobb's Journal
<http://www.ddj.com/topics/xml/>

eWeek Developer/Web Services
<http://www.eweek.com/category2/0,4148,1237895,00.asp>

InfoWorld Web Services
http://www.infoworld.com/techindex/websvcs_1.html

Application Development Trends Magazine
<http://www.adtmag.com/topic.asp?k=Web+Services>

Base64 Decoding

Some API calls retrieve data as MIME base-64-encoded ASCII strings. For example, XSL stylesheets that can be used when listing with Item Specifics and Pre-filled Item Information are returned in this format.

Base64 (for Java)
SourceForge
<http://iharder.sourceforge.net/base64>

How to Base64
KBCafe
<http://www.kbcafe.com/articles/HowTo.Base64.pdf>

HTML Forms

Introduction to Forms

Microsoft Corporation

<http://msdn.microsoft.com/workshop/author/forms/formsoverview.asp>

MSDN Library DHTML Methods

Microsoft Corporation

<http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/objects/form.asp>

Internet Explorer WebBrowser Control

WebBrowser Control

Microsoft Corporation

http://msdn.microsoft.com/workshop/browser/webbrowser/browser_control_node_entry.asp

IHTMLDocument2 Interface

Microsoft Corporation

<http://msdn.microsoft.com/library/default.asp?url=/workshop/browser/mshtml/reference/ifaces/document2/document2.asp>

JavaScript call from C++

The Code Project

<http://www.codeproject.com/com/jscalls.asp>

Handling HTML Element Events in MFC applications

The Code Project

<http://codeproject.com/shell/dlgdhtmlevents.asp>

Regular Expressions

java.util.regex.Pattern

(Provides a complete list of the valid constructs)

Sun Microsystems, Inc.

<http://java.sun.com/j2se/1.4.1/docs/api/java/util/regex/Pattern.html>

perlre

(Perl Regular Expressions)

Perldoc

<http://www.perldoc.com/perl5.8.0/pod/perlre.html>

The GRETA Regular Expression Template Archive

(For C++ development)

Microsoft Corporation

<http://research.microsoft.com/projects/greta/>

Other Resources

Dates and Time, ISO 8601

International Organization for Standardization

<http://www.iso.org/iso/en/prods-services/popstds/datesandtime.html>

ISO 8601 Date and Time Formats

The World Wide Web Consortium (W3C)

<http://www.w3.org/TR/xmlschema-2/#isoformats>

English Country Names and Code Elements, ISO 3166

International Organization for Standardization

<http://www.iso.ch/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html>

Appendix B Data Types

The API uses a number of standard simple data types and defines a few additional core data types.

- “[Summary of Data Types](#)” on page 959
- “[Time Values](#)” on page 960

Summary of Data Types

This section covers the following topics:

- “[eBay Namespace](#)” on page 959
- “[eBay Types](#)” on page 959
- “[XML Schema Simple Types](#)” on page 959

eBay Namespace

The schema identifies one eBay namespace: `xmlns:ebl="urn:ebay:apis:eBLBaseComponents"`. When you are making calls you only need specify the namespace on the “Request” tag in the body of the call.

eBay Types

eBay defines a number of complex and simple types in the schema, such as [ItemType](#) and [ListingDurationCodeType](#). See the following topic for a link to the documentation of complex and simple types: “[Schema Reference Documentation](#)” on page 967.

Also see “[Overview of the API Schema](#)” on page 75.

XML Schema Simple Types

A number of elements in the schema, such as [CategoryID](#), are XML Schema simple types (e.g., `string`).

Simple Schema Types

<http://developer.ebay.com/DevZone/XML/docs/Reference/eBay/types/simpleTypes.html>

Time Values

This section covers the following topics:

- “Why Do Time Values Seem Inconsistent with the Web Site?” on page 960
- “About GMT” on page 960
- “Converting Between UTC/GMT and Local Time” on page 960
- “Converting Between Pacific Time (PST/PDT) and a Local Time” on page 962

Why Do Time Values Seem Inconsistent with the Web Site?

Depending on your location and your development environment, date-time values returned from eBay may appear to be different from the time stamps you see on the eBay.com Web site and the Sandbox Web site.

The eBay.com Web site displays time values in a local time zone or in Pacific Time. For example, the US site uses Pacific Time. However, in most cases, the eBay Platform time-stamps all events and transactions in Universal Coordinated Time (UTC), also known as Greenwich Mean Time (GMT). So, unless otherwise specified for a particular field or call (e.g., `GetSearchResults` uses Pacific Time with compatibility levels lower than 387), all date-time values returned from eBay are in GMT (specifically, the Zulu format), not in the time zone displayed on the Web site.

Hours are in the 24-hour format (e.g., 2:00:00pm is shown as 14:00:00).

ISO 8601 Date and Time Formats
The World Wide Web Consortium (W3C)
<http://www.w3.org/TR/xmlschema-2/#isoformats>

For simplicity, we refer to UTC/GMT/Zulu time “GMT” in this documentation.

About GMT

By international convention, GMT is used as the basis for determining the official time in all time zones around the world. GMT express the time of day on Earth at 0° longitude (the Greenwich Meridian). Historically, GMT is associated with Greenwich, England. Unlike many local time zones, GMT does not switch to daylight-saving time in the spring.

Converting Between UTC/GMT and Local Time

If you prefer to display or work with date-time values in a particular time zone, your application needs to handle the conversion between GMT and that time zone.

Using Third-Party Libraries

Some programming languages include classes, functions, modules, or other constructs for converting between a local time and GMT (or times in any two zones) and adjusting for daylight-saving time.

Some development environments automatically convert SOAP `dateTime` values to your local system time. In this case, you would only need to convert a time value if you want to display a value other than your local system time, or if you need to convert time values in order to work with the XML API.

Performing the Conversion Yourself

To program the conversion to or from GMT by using your own algorithm, you need to know the GMT offset and daylight-saving time behavior in the local time zone. (Two locations with the same GMT offset might not use the same daylight-saving time rules.) The offset is the difference between the current time in the local time zone and GMT. GMT does not switch to daylight-saving time, so if the local time zone does, the offset will change twice a year. Depending on the country the user is in, daylight-saving time starts in March or April and lasts through September or October.

Information about GMT offsets for different time zones is widely available. For example, you can search for "Time Zone Guide GMT" on the Internet. Your computer's clock application may also list GMT offsets for different time zones.

West of the Greenwich Meridian: If the local time zone is west of the Greenwich Meridian, the offset is negative. For example, when it is 17:00:00 GMT (5:00 PM) GMT, it is 09:00:00 PST (9:00 AM Pacific Standard Time) in California, US during winter. So, the GMT offset for PST is -8:00.

To convert from GMT to PST and back, you use these formulas:

$$\begin{aligned} \text{PST} &= \text{GMT} - 8 \text{ hours} \\ \text{GMT} &= \text{PST} + 8 \text{ hours} \end{aligned}$$

$$\begin{aligned} \text{PST} &= 17:00:00 - 8:00:00 = 09:00:00 \\ \text{GMT} &= 09:00:00 + 8:00:00 = 17:00:00 \end{aligned}$$

During Pacific Daylight Time (PDT), the GMT offset for PDT is -7:00 (-8 + 1):

$$\begin{aligned} \text{PDT} &= \text{GMT} - 7 \text{ hours} \\ \text{GMT} &= \text{PDT} + 7 \text{ hours} \end{aligned}$$

$$\begin{aligned} \text{PDT} &= 17:00:00 - 7:00:00 = 10:00:00 \\ \text{GMT} &= 10:00:00 + 7:00:00 = 17:00:00 \end{aligned}$$

East of the Greenwich Meridian: If the local time zone is east of the Greenwich Meridian, the offset is positive. For example, when it is 17:00:00 (5:00 PM) GMT, it is 03:00:00 AEST (3:00 AM Australian Eastern Standard Time) the next morning in Canberra, Australia. So, the GMT offset for AEST is +10:00.

To convert from GMT to AEST and back, you use these formulas:

$$\begin{aligned} \text{AEST} &= \text{GMT} + 10 \text{ hours} \\ \text{GMT} &= \text{AEST} - 10 \text{ hours} \end{aligned}$$

$$\begin{aligned} \text{AEST} &= 17:00:00 + 10:00:00 = 03:00:00 \\ \text{GMT} &= 03:00:00 - 10:00:00 = 17:00:00 \end{aligned}$$

During Australian Eastern Daylight Time (AEDT), the GMT offset for AEDT is +11:00 (+10 + 1):

$$\begin{aligned} \text{AEST} &= \text{GMT} + 11 \text{ hours} \\ \text{GMT} &= \text{AEST} - 11 \text{ hours} \end{aligned}$$

$$\begin{aligned} \text{AEST} &= 17:00:00 + 11:00:00 = 04:00:00 \\ \text{GMT} &= 04:00:00 - 11:00:00 = 17:00:00 \end{aligned}$$

Converting Between Pacific Time (PST/PDT) and a Local Time

If you need to convert Pacific Time values to time values for another time zone, you can use one of these approaches:

- Use the formulas described in “[Converting Between UTC/GMT and Local Time](#)” on page 960 to convert from Pacific Time to GMT and then from GMT to the corresponding time in the local time zone.
- Determine the offset between Pacific Time and the local time and convert the value in one step.

Appendix C

Compatible Application Check for Specific Calls

When using the Production environment, certain calls have additional Compatible Application Check Requirements. See “[Compatible Application Check](#)” on page 67 for general information about Compatible Application Check Requirements.

[Table C-1](#) lists the Compatible Application Check Requirements for specific calls.

Table C-1 Compatible Application Check Requirements for Specific Calls

Call Name	Description
GetCategories	Execute GetCategories with no detail level and set the <code>ViewAllNodes</code> field to false in order to check the version of the eBay category hierarchy. This should be done at least once a day to ensure the locally cached category hierarchy is up to date. If the version has changed from the version you have stored locally, then execute GetCategories and set the <code>ViewAllNodes</code> field to true to download the updates to the hierarchy. Should be done only when necessary (i.e., a new category tree version is indicated).
GetCategoryListings	When collecting new listings, the application should limit GetCategoryListings request frequency to no more than once every 30 minutes.
GeteBayDetails	As described in the <i>API Logo Usage Requirements</i> and your <i>API License Agreement</i> , if your application publicly displays eBay content, you must display the eBay Content Logo in conjunction with the eBay content. To retrieve the most current version of this logo, you must call GeteBayDetails at least once a day, and cache the logo locally. For more information on logo usage, please see http://developer.ebay.com/join/licenses/apilogousage on the Developer Web site. GeteBayDetails returns the eBay Content Logo URLs with other (optional) URL details. You can choose from one of three available sizes (small, medium, or large).
GetFeedback	When retrieving feedback, the application should only retrieve new feedback records. (A “new” feedback record is a record that your application has not retrieved previously.) Older feedback records should be cached. That is, the application should cache feedback records retrieved, and also store the <code>FeedbackDetailItemTotal</code> value, so that subsequent calls for already-retrieved records are unnecessary (GetFeedback returns the most recent records first). For additional information, see https://ebay.custhelp.com/cgi-bin/ebay.cfg/php/enduser/std_adp.php?p_faqid=411 .
GetItem	Only use GetItem when you cannot use GetSellerEvents , because GetSellerEvents is a more efficient call than GetItem . For example, for any given listing, you should only need to use GetItem when you have not yet stored the details of a particular listing locally. After you have stored the details locally, you should use GetSellerEvents when you want to know whether the status of any of a seller’s listings has changed.
GetSellerEvents	When using GetSellerEvents , specify a time window (using <code>EndTimeFrom</code> and <code>EndTimeTo</code> , or <code>ModTimeFrom</code> and <code>ModTimeTo</code>) that matches the window of your calling frequency. For example, if you call GetSellerEvents every 30 minutes, use an <code>EndTime</code> window that also is 30 minutes.

Table C-1 Compatible Application Check Requirements for Specific Calls

Call Name	Description
GetSellerList	After you have imported a seller's items to your local application for the first time, only use GetSellerList if you cannot use GetSellerEvents . GetSellerEvents is a more efficient call than GetSellerList .

Appendix D

Current Compatibility Levels

This appendix has been replaced by the new [Versioning Summary](#) page in the *eBay Trading API Call Reference*.

Appendix E

Error Message Strings

The *Error Message Strings* documentation lists all of the error messages that could be encountered when invoking *eBay Web Services*. The information for each error includes: error code, error severity, short error string, and long error string. All error messages are documented only in English, but the error codes can be used as an index for a structure, array, or string table to supply messages in whatever language is desired.

Please note that error 10007 (“Internal error to the application”) indicates an error on the eBay server side, not an error in your application.

Information about error message strings is provided as an HTML document.

If you are reading the PDF version of this guide, you can access the Error Message Strings documentation via the online documentation on the eBay Developer Program site.

[Error Message Strings](#)

In addition, the *Top error codes returned per call* document lists typical errors returned for some of the most commonly used calls in production. These calls may return additional errors not shown in the list.

[Top Error Codes Returned Per Call](#)

Schema Reference Documentation

Detailed information about each property in the Trading API is provided in the Trading API Call Reference.

The Trading API Call Reference describes the call-specific input and output fields for Trading API calls. The reference is generated using the eBay Trading API schema.

[Trading API Call Reference](#)

The Trading API Call Reference also provides a “Type” view of the schema (accessible by clicking “Type Index” at the top of any call page).

The Type view, also called the “Type documentation” and “Type Reference,” tells you all the fields defined on a type.

This view can be useful if you are using an eBay SDK and you want to map the types in the schema to classes in the SDK. However, note that some calls only use a subset of the fields defined on a type. (That’s why you usually need to look at the Trading API Call Reference.) The Type Reference is at the following location:



[Type Reference](#)

Note: The following old-style documents are frozen at schema version 537 and will no longer be updated:

[Schema Documentation](#)

[Visual Schema Reference](#)

As of October 31, 2007, these old-style documents are no longer supported. Please use the Type reference instead.

The latest eBay schema is in the following locations (see “[Where to Find the eBay Schema Files](#)” on page 51):

ebaySvc.xsd

<http://developer.ebay.com/webservices/latest/ebaySvc.xsd>

ebaySvc.wsdl

<http://developer.ebay.com/webservices/latest/ebaySvc.wsdl>

Part 3

Index

Index

969

Index

A

About Catalog Products 143
abstract types
 in the schema 75
access rules 802
account informaton
 eBay Developers Program 36
accounts
 retrieving 521
AddDispute
 defined 771
AddDisputeResponse 467
 defined 775
AddFixedPriceItem 781
AddItem 782
 defined 843
AddItems 783
AddToItemDescription
 how-to 388
AddToItemDescriptionCall
 how-to 388
AddTransactionConfirmationItem 791
AdType
 international differences 604
adult items 238
Affiliate Tracking 155
Agreement, License 49
amounts
 as core components 78
anonymous best offers 228
Apache Axis, downloading 50
API calls, overview of 770
AppId
 defined 71
appid
 gateway URL and 89
application
 Selling Manager 397
application-level errors
 see *errors*
applications
 building flexibility into 58
 compatible application check
 deploying
 see *certification*
 developing 56
 concepts 56
 gallery list, defined 49
 keys
 defined 71
 AppID 89
 listing, defined 48
 seller, defined 49

testing 56, 61
 concepts 56
 types you can build 48
Ask Seller a Question 404
AskSellerQuestion notification 636
AttrAboutItemSpecifics 141
attributes
 category-based pre-fills 163
 return policy 223
 vs. custom Item Specifics 481
AuctionCheckoutComplete notification 638
auctions
 binding 367
 nonbinding 367
 private auctions 237
 types
 Auctions 137
 Basic Fixed-Price 138
 Buy It Now 137
 Real Estate 137
 Second Chance Offer 138
Australia
 eBay Motors 573
AuthCert
 defined 71
authentication
 defined 101
 by password 103
 by token 101
 sign-in redirects 125
Authentication & Authorization 117
automated sandbox test users 63

B

backward compatibility
 see also *compatibility levels*
 interoperability and 86
base components 79
Basic Fixed-Price listing type 138
Best Offer feature
 defined 225
 enabling 225
Best Offer Only format 562
best offers
 accepting 225
 anonymous 228
 auto-decline 225
 counteroffers 225
 declining 225
 expiration 225
best practices
 compatibility levels and 58
 for specific calls 58

- overview 58
 - BestOffer notification** 642, 646
 - BestOfferRejected notification** 644
 - bidders**
 - defined 148
 - learning about 393
 - BidReceived notification** 651
 - binding**
 - auctions 367
 - clause, in WSDL 91
 - boolean values**
 - usage in schema 959
 - building applications**
 - best practices 58
 - common types 48
 - BulkDataExchangeJobCompleted notification** 656
 - business features**
 - international differences 591
 - Buyer Disputes** 476
 - buyer payment, understanding** 420
 - Buyer Protection Program** 196
 - buyers**
 - defined 148
 - buying guides**
 - in category listings 500
 - in search results 499
 - Buy It Now**
 - defined 137
- C**
- calculated shipping** 312
 - callname**
 - gateway URL and 89
 - calls**
 - executing your first 55
 - following best practices 58
 - list of available 770
 - standard data 92
 - cardinality**
 - interoperability and 86
 - minOccurs 81
 - categories** 158
 - defined 138
 - attribute defaults for 163
 - finding items by 501
 - mapping 161
 - mappings 163
 - reserve price 169
 - retrieving 139, 158
 - return policy support 225
 - suggested 171
 - US eBay Motors 559
 - using 139
 - category mapping** 161, 163
 - category-based attribute pre-fills** 163
 - CertId**
 - defined 71
 - certification**
 - how-to 70
- requirements**
 - error handling 68, 69
 - pagination 69
 - simultaneous requests 69
 - timeline 71
 - changes**
 - finding out about 36
 - how eBay handles 58
 - charitable items**
 - other ways to list 236
 - see *eBay Giving Works*
 - checkout**
 - defined 407
 - Combined Payment 183
 - Immediate Payment 187
 - Checkout notification** 658
 - CheckoutBuyerRequestsTotal notification** 665
 - Chinese auctions**
 - Buy It Now option 137
 - claim**
 - retrieving
 - getUserCases, getEBPCaseDetail 446
 - classified ad items, working with** 366
 - client environments, supported** 50
 - code lists** 80
 - defined 82
 - annotations 84
 - limitations of 84
 - CodeType**
 - see *code lists*
 - Combined Payment** 183
 - combined payment**
 - packaging/handling cost 361
 - shipping insurance 361
 - compatibility levels**
 - defined 71
 - current list of 965
 - supporting 71
 - compatible application check**
 - requirements
 - defined 67
 - for specific calls 963
 - completing the sale** 415
 - components**
 - base 79
 - code lists 82
 - common features 75
 - core 78
 - error types 108
 - predefined values for 80
 - concrete message types** 76
 - condition**
 - category differences 220
 - new or used 218
 - condition description** 221
 - ConditionDescription** 221
 - ConditionID** 218
 - configuring output** 59
 - contacting**
 - Developer Relations 39
 - Developer Technical Support 38

controlling response sizes 100
conventions
 concrete message names 76
 schema documentation 84
 typographic, in this guide 42
core components
 defined 78
 amounts 78
 measures 78
correlation ID 76
counteroffers
 buyer 226
 expiration 226
 retracting 226
 rules and restrictions 226
custom Item Specifics 481
CustomCode
 defined 82
 troubleshooting 82

D

dashboard, seller 520
data
 controlling the amount returned 100
 controlling with detail levels 98
 standard for all calls 92
 validation 80
data types
 list of standard 959
databases
 choosing 53
 third-party links 53
date values
 usage in schema 959
decimal values
 usage in schema 959
default attributes
 category-based 163
default values
 CustomCode 84
deploying applications
 see *certification*
deposit
 US eBay Motors listings 568
describing items 206
description
 condition 221
Description Templates 253
detail levels
 defined 98
 GetItem 895
 GetWantItNowPost 890
Developer Forums 36
Developer Relations, contacting 39
Developer Technical Support, contacting 38
DevId
 defined 71
discount pricing
 display rules 505
 minimum advertised price 505

strikethrough pricing 505
DispatchTimeMax 181
display rules
 minimum advertised price 505
 strikethrough pricing 505
dispute
 adding a comment 466
 closing 467
 getting information 469, 470
 retrieving
 getUserName, GetDispute 446
 reversing
 SellerReverseDispute 470
DisputeActivity
 completing transaction 468
DisputeID 466
Disputes
 overview 441
DisputeState 466
distance search
see proximity search
documentation
 code lists 84
 schema
 diagram conventions 43
 downloading 40
 location of 39, 40
this guide
 about 39
 diagram conventions 43
 downloading 39
 intended audience 40
 organization of 40
 typographic conventions in 42
domestic handling 181
domestic shipping 312
downloading
 Apache Axis 50
 Microsoft .NET Framework 50
 schema documentation 40
this guide 39
duration
 international differences 596

E

eBay
 learning to sell on 37
 official time 76, 960
SDK
 for Java 46
 for Windows 46
 Web services 45
eBay Buyer Protection (eBP) 666
eBay Buyer Protection notifications 666
eBay Classified Ads
 working with 366
eBay Developer News 36
eBay Germany
 parts compatibility 282
 specifying fitment 282

eBay Giving Works
 API support for 234
 benefits of using 236
 retrieving listings 234

eBay Motors
 Australia 573
 VIN for Australia 573

eBay Motors (US)
 defined 153

eBay Motors (DE)
 cross-promoting on mobile.de 572

eBay Motors (US)
 deposit 568
 Immediate Payment 569
 items in eBay Stores 543, 570
 limited warranty 570
 listing items 559
 Manufacturer and Model 567
 parts compatibility 282
 payment methods 568
 searching for listings on 571
 specifying fitment 282
 subtitle 568
 title 568
 working with 559

eBay MotorsGermany 572

eBay Partner Network 155

eBay Real Estate (US) 137
 working with 367

eBay Return 746

eBay sites 149

eBay Stores
 defined 154
 Sandbox 67
 special fields 543
 Store details 524
 Store owners 148
 vacation hold for 542
 working with 534

eBayAPIInterface 91

eBayAPIInterfaceService 91

eBayAPISoapBinding 91

eBayAuthToken 101

eBaySvc.wsdl
 generating interfaces and stubs 51
 location of 51

eBayTypes.wsdl
 generating stubs 51
 location of 51

elements
 international differences 590

Email
 PayPal Sandbox 201, 203

end of auction processing 415

EndFixedPriceItem 795

EndOfAuction notification 670

error language 97

errors
 application-level
 certification requirements 69

application-level errors
 defined 108
 due to category IDs 163
 error handling
 application customizations 113
 certification requirements 68
 compatible application check requirements 115
 explained 107
 getting help 115
 infrastructure-level
 defined 111
 certification requirements 68
 localizing 97
 other errors
 defined 113
 types 108
 why some fields are ignored 80

exceptions
 see *errors*

executing
 calls 55

F

fast delivery 181

faults
 see *errors*

feedback
 about users
 defined 145
 end of auction processing 426
 how-to 432
 technical
 Developer Forums 36
 feature requests 35

Feedback notification 675

FeedbackLeft notification 677

FeedbackReceived notification 679

FeedbackStarChanged notification 681

fees
 listing 59

fields
 international differences 590
 optional
 explained 81
 required
 explained 81
 standard input 92
 why some are ignored 80

File Exchange 574

filtering data 100

fitment
 see *parts compatibility*

fixed-price listings
 defined 138
 multi-item variations 264
 variations 264

FixedPriceTransaction notification 684

flat shipping 312

float values
 usage in schema 959

Foreword 35
forums 36
Free Gallery 215
free shipping 394
frequency
 of releases 85

G

Gallery 213, 215
 Gallery Featured 213
 pictures 213
Gallery Featured 213, 215
gallery list tool, defined 49
Gallery Plus 215
Gateway URL 87
generating
 interfaces and stubs from WSDL 51
Germany
 eBay Motors 572
Get It Fast 181
GetAccount 799
GetAdFormatLeads 800
GetAllBidders
 options 393
GetAllBiddersCall 801
GetApiAccessRules
 defined 802
GetCategories
 certification requirements 963
 how-to 158
GetCategory2CS 806
GetCategory2FinanceOffers 807
GetCategoryMappings 810
 using 161
GetCategorySpecifics 811
GetChallengeToken 812
GetCharities
 defined 813
GetClientAlertsAuthToken 824
GetContextualKeywords 825
GetCrossPromotions 826
GetDescriptionTemplates 827
GetDispute 469
GetItem 833
 certification requirements 963
 detail levels 895
GetItemsAwaitingFeedback 834
GetItFast 181
GetNotificationUsage
 defined 849
GetPromotionalSaleDetails 854
GetPromotionRules 855
GetSellerDashboardCall
 defined 857
GetSellerEvents
 certification requirements 963
GetSellerList 859
 certification requirements 963
GetSellerPaymentsCall
 defined 860

GetShippingDiscountProfiles 867
GetStore 868
GetStoreCategoryUpdateStatus 869
GetStoreCustomPage 936
GetStoreOptions 870
 defined 871
GetStorePreferences 872
GetSuggestedCategories 171, 873
 query string 172
getting certified
 see certification
 GetUserDisputes 470
 GetUserPreferences 879
GetVeROReasonCodeDetails
 defined 880
GetVeROReportStatus
 defined 884
GetWantItNowPost
 defined 889
 detail levels 890
GetWantItNowSearchResults
 defined 893
GMT
 converting to and from 960

H

Half.com 574
 defined 574
 commissions 575
 defined 155
 listing items to 574
 listing workflow 574
 pay periods 576
 payments 575
 refunds 575
 registration 575
 sellers 575
 working with 574
handling errors
 see errors
hardware
 requirements 53
help
 where to get 35
HitCounter
 international differences 595
Holds on PayPal Payments 421
how to
 execute a call 55
HTTP 200 response 108
HTTP 500 error 111

ignored fields
 why some fields are ignored 80
Immediate Payment
 defined 187
 testing 188
 US eBay Motors 569

- Informational alerts** 631
- infrastructure-level errors**
 - see *errors*
- input fields, standard** 92
- Instant Payment Notification simulator** 204
- int values**
 - usage in schema 959
- integer values**
 - usage in schema 959
- international**
 - field differences 590
- international eBay sites** 150
- international shipping** 312
- interoperability** 86
- invoking**
 - eBay Web Services 74
- IPN (Instant Payment Notification) simulator** 204
- ISO**
 - 3166, Country Names and Codes 958
 - 8601, Dates and Time 958
 - third-party resources 958
- IssueRefund**
 - defined 900
- item condition**
 - category differences 220
- Item Specifics**
 - defined 139
 - custom 481
 - item condition 218
 - return policy 223
 - variations 264
- ItemAddedToWatchList notification** 691
- ItemClosed notification** 694
- ItemListed notification** 699
- ItemLost notification** 703
- ItemMarkedPaid notification** 707
- ItemMarkedShipped notification** 711
- ItemNotReceived notification** 685
- ItemRemovedFromWatchList notification** 718
- ItemRevised notification** 722
- items**
 - defined 136
 - add text to description 388
 - adult 238
 - browsing sellers' 502
 - charitable
 - classified ad 366
 - condition 218
 - cross-promoting on mobile.de 572
 - describing 206
 - Description Templates 253
 - distance-based search 603
 - eBay Motors (US) 559
 - eBay Stores 534
 - special fields 543
 - US eBay Motors items in 543, 570
 - ending early 389
 - finding by category 501
 - Gallery 213, 215
 - Gallery Featured 213
 - Half.com 574
- listing new** 176
- new and used** 218
- offering a return policy** 223
- pictures** 206
- private auctions** 237
- proximity search** 603
- real estate (US)** 367
- re-listing** 376
- retrieving** 502
 - Item Specifics 504
 - one 504
 - Pre-filled Item Information 504
- revising** 381
- searching for** 498
- Second Chance Offers** 426
- seller events** 391
- shipping items** 424
- SKU numbers** 239
- Items Near Me** 498
- ItemUnsold notification** 725, 729, 762
- ItemWon notification** 733

K

- Keys**
 - Production
 - defined 71
- Sandbox**
 - defined 71
- Knowledge Base** 36
- known issues page, location of** 37
- KType vehicle number** 282
 - specifying parts compatibility 295

L

- language, error** 97
- lead generation**
 - Best Offer Only 562
- levels, compatibility**
 - see *compatibility levels*
- License Agreement, location of** 49
- limited warranty**
 - US eBay Motors 570
- limits**
 - string lengths 82
 - see also *compatible application check*
- listing an item** 176
- listing applications, defined** 48
- Listing Designer** 253
- Listing policies** 174
- ListingDuration**
 - international differences 596
- Auctions**
 - listings**
 - defined 137
- listings**
 - defined 136
 - add text to description 388
 - adult 238
 - browsing sellers' 502

- classified ad 366
 - condition 218
 - creating 176
 - cross-promoting on mobile.de 572
 - Description Templates 253
 - eBay Motors (US) 559
 - eBay Stores 534
 - ending early 389
 - fees 59
 - finding by category 501
 - Gallery 213, 215
 - Gallery Featured 213
 - Half.com 574
 - new and used items 218
 - offering a return policy 223
 - pictures 206
 - private auctions 237
 - real estate (US) 367
 - re-listing 376
 - removing subtitle 380
 - retrieving one 504
 - revising 381
 - searching for 498
 - Second Chance Offers 426
 - seller events 391
 - types 136
 - Auctions 137
 - Basic Fixed-Price 138
 - Buy It Now 137
 - eBay Real Estate (US) 137
 - Second Chance Offer 138
 - with Item Specifics
 - retrieving 504
 - with Pre-filled Item Information
 - retrieving 504
- ListingType**
- international differences 604
- live applications**
- see *Production environment*
- local listings** 562
- Local Market listings** 562
- M**
- maintaining**
- categories 163
 - category data 161
- managing item shipping costs** 312
- managing shipping** 312
- managing user information** 513
- Manual test account, PayPal Sandbox** 202
- Manufacturer attribute**
- US eBay Motors listings 567
- mapping**
- categories 163
 - old and active categories 161
- Markdown Manager** 394
- measures**
- as core components 78
- Member Forums** 36
- message ID** 76, 96
- messages**
- types of 75
- Microsoft .NET Framework**
- downloading 50
- minimum advertised price** 505
- MinimumReservePrice** 169
- minOccurs, explained** 81
- mobile.de site**
- defined 572
 - cross-promoting vehicles on 572
- Model attribute**
- US eBay Motors listings 567
- modifying**
- eBay Developers Program membership 36
- Motors**
- see *eBay Motors (US)*
- multi-item variations** 264
- multiplicity**
- interoperability and 86
 - minOccurs 81
- My eBay**
- replicating 525, 601
 - user preferences on 517
- MyMessages notification** 737
- N**
- new item condition** 218
- newsletter** 36
- nonbinding auctions** 367
- non-paying bidders** 424
- Notifications**
- AskSellerQuestion 636
 - AuctionCheckoutComplete 638
 - BestOffer 642, 646
 - BestOfferRejected 644
 - BidReceived 651
 - BulkDataExchangeJobCompleted 656
 - Checkout 658
 - CheckoutBuyerRequestsTotal 665
 - eBay Buyer Protection 666
 - EndOfAuction 670
 - Feedback 675
 - FeedbackLeft 677
 - FeedbackReceived 679
 - FeedbackStarChanged 681
 - FixedPriceTransaction 684
 - ItemAddedToWatchList 691
 - ItemClosed 694
 - ItemListed 699
 - ItemLost 703
 - ItemMarkedPaid 707
 - ItemMarkedShipped 711
 - ItemNotReceived 685
 - ItemRemovedFromWatchList 718
 - ItemRevised 722
 - ItemUnsold 725, 729, 762
 - ItemWon 733
 - MyMessages 737
 - OutBid 742
 - RemoveFromBidGroup 715

SecondChanceOffer 750
 Seller Payouts 758
 WatchedItemEndingSoon 663, 763

O

object model 115
 obsolete
 categories 163
 optional fields
 explained 81
 orders
 see *Combined Payment*
 ORPA 169
 OutBid notification 742
 OutputSelector field 59
 overview
 fees 59
 product 45
 schema 75

P

packaging handling cost 361
 pagination
 defined 100
 certification requirements 69
 GetSellerList 503
 parts compatibility
 by application 282
 by specification 282
 defined 282
 using KType number 282
 password 62
 pay periods
 Half.com 576
 payload version 95
 payment 420
 payment method
 changing 384
 payment methods
 changing 378
 US eBay Motors 568
 PaymentMethod
 international differences 185
 payments
 Half.com sales 575
 PayPal
 defined 153
 buyer payment, and 420
 Buyer Protection, and 196
 Immediate Payment 187
 integrating with eBay applications 197
 overview 195
 Sandbox 199
 PayPal Buyer Protection 444
 PDT (Pacific Daylight Time)
 see *time values*
 performance
 certification requirements 67
 pictures, item 206

planning
 certification 57
 development process 56
 Platform Notifications
 best practices 59
 Policies
 listing 174
 polling frequency limits 69
 polymorphism
 in the schema 86
 postal codes 603
 PowerSeller status 520
 Preconfigured test account, PayPal Sandbox 200
 predefined values 80, 82
 preferences, user
 retrieving 517
 setting 517
 pre-filled attributes
 category-based 163
 Pre-filled Item Information
 defined 139
 re-listing items with 379
 revising items with 386
 price discount 394
 private auctions 237
 product
 about this 45
 Selling Manager 397
 Production environment
 defined 149
 checklist for going live 67
 gateway URI 87
 keys, getting 71
 see also *compatible application check*
 promotional sales 394, 610
 properties
 international differences 590
 proximity search
 supporting in listings 179, 603
 PST (Pacific Standard Time)
 see *time values*
 public messages 404

Q

query strings
 gateway URL 89, 90
 GetSuggestedCategories 172

R

real estate items (US), working with 367
 Reducing reserve price 388
 refunds
 Half.com sales 575
 Registration Date
 and VIN 573
 release
 cycle 85
 notes, location of 37
 supported use cases 47

- versions 84
 - RelistFixedPriceItem** 905
 - re-listing an item** 376
 - with Pre-filled Item Information 379
 - relisting items**
 - changing payment method 378
 - RemoveFromBidGroup notification** 715
 - RemoveFromWatchList** 907
 - requests**
 - routing 87
 - routing to Production 87
 - routing to the Sandbox 87
 - required fields, explained** 81
 - requirements**
 - certification 57
 - error handling 68
 - see also *certification*
 - compatible application check
 - for specific calls 963
 - hardware 53
 - system 50
 - Reserve Price** 388
 - reserve price** 169
 - ReservePriceInclusive** 169
 - reset PayPal Sandbox test account** 202
 - Resolution Case Management API** 446
 - resources**
 - developer relations 39
 - developer support 38
 - free technical 35
 - learning to sell on eBay 37
 - third-party 953
 - databases 53
 - Web services 953
 - RespondToBestOffer**
 - defined 908
 - RespondToWantItNowPost**
 - defined 913
 - retrieving account data** 521
 - retrieving items** 502
 - retrieving user preferences** 517
 - retrieving users** 513
 - return policy**
 - attributes 223
 - payment instructions 223
 - Review Your Purchase** 411
 - revised items**
 - changing payment method 384
 - ReviseFixedPriceItem**
 - defined 917
 - ReviseMyMessages** 919
 - ReviseMyMessagesFolders** 920
 - revising an item** 381
 - with Pre-filled Item Information 386
 - roles, user** 147
 - routing the request** 87
 - rules**
 - certification requirements 57
 - License Agreement and Usage 49
 - see also *compatible application check*
 - runame** 125
 - ruparams** 125
- S**
- sales**
 - free shipping 394
 - price discount 394
 - site differences 610
 - sample code**
 - executing your first call 55
 - Sandbox**
 - PayPal 199
 - sandbox**
 - auth token 62
 - automated test users 63
 - password 62
 - username 62
 - Sandbox environment**
 - defined 149
 - automated test users 63
 - creating Stores 67
 - creating test users 63
 - gateway URI 87
 - keys, getting 71
 - using the Sandbox 62
 - schema**
 - common components 75
 - documentation
 - diagram conventions 43
 - location of 40
 - interoperability considerations 86
 - overview 75
 - version 95
 - SDKs**
 - for Java 46
 - for Windows 46
 - Search**
 - international differences 605
 - searching**
 - by distance 179, 603
 - by proximity 179, 603
 - for buying guides 499
 - for items 498
 - for mobile.de-enabled items 572
 - for US eBay Motors listings 571
 - searching for items** 498
 - Second Chance Offer**
 - listing type, defined 138
 - Second Chance Offers** 426
 - second chance offers** 393
 - SecondChanceOffer notification** 750
 - security** 101
 - seller** 226
 - seller applications, defined** 49
 - Seller Dashboard** 520
 - seller events** 391
 - Seller Payout notifications** 758
 - SellerLevel**
 - international differences 607
 - SellerReverseDispute**
 - defined 922

- sellers**
 - defined 148
 - Half.com 575
 - Store owners 148
- Selling Manager** 397
- selling on eBay**
 - learning how 37
- SendInvoice** 927
- service definition interface** 91
- SetNotificationPreferences** 929
- SetPromotionalSale** 930
- SetPromotionalSaleListings** 931
- SetPromotionRules** 932
- SetShippingDiscountProfiles** 933
- SetStore** 934
- SetStoreCategories** 935
- SetStorePreferences** 937
- setting user preferences** 517
- SetUserNotes** 939
- SetUserPreferences**
 - defined 940
- shipping**
 - key calls 352
 - shipping items
 - PayPal, and 424
 - US eBay Motors items 559
- shipping costs** 312
- shipping insurance** 312, 361
- shipping regions** 312
- shipping services** 312
- ShippingOption**
 - international differences 609
- ShippingRegion**
 - international differences 609
- Shopping.com**
 - defined 154
- Short Term Service Agreement**
 - US eBay Motors 570
- simultaneous requests**
 - see *certification requirements*
- site ID**
 - eBayMotors 559
 - gateway URL and 89
- sites**
 - eBay Classified Ads 366
 - eBay Real Estate (US) 367
 - eBay Stores 534
 - international differences 590
 - return policy support 225
 - specifying in requests 93
 - US eBay Motors 559
 - working with special 557
- SMS** 634
- SOAP**
 - faults
 - see *errors*
 - Gateway URL 87
- SOAP API**
 - defined 45
- specifying the target eBay site** 93
- standard input fields** 92
- standards for all calls** 92
- Store preferences**
 - getting 542
 - managing 542
 - setting 542
- Stores**
 - editing 536
 - managing 534
 - managing categories 539
 - managing custom pages 540
 - retrieving a store 537
 - submitting changes 537
 - see *eBay Stores*
- stores**
 - eBay Motors items in 559
- Stores sellers**
 - defined 534
 - special fields for 543
- storing**
 - data in a database 53
- strikethrough pricing** 505
- string values**
 - usage in schema 959
- strings**
 - length restrictions 82
- subtitle**
 - removing 380
 - US eBay Motors listings 568
- suggested categories** 171
- support**
 - contacting (fee-based) 38
 - technical 35
 - fee-based 38
 - free resources 35
- supported**
 - client environments 50
 - use cases in this release 47
- supporting**
 - compatibility levels 71
- System Announcements Board** 37
- system requirements** 50

T

- tax fields**
 - international differences 614
- technical support**
 - contacting (fee-based) 38
- Test accounts in PayPal Sandbox** 200
- test users**
 - Sandbox 63
- testing**
 - applications 61
 - automated test users 63
 - creating a test user 63
 - listings in eBay Stores 67
- third-party resources** 953
 - database links 53
 - databases 53
 - ISO 3166, Country Names and Codes 958
 - ISO 8601, Dates and Time 958

Web services 953
time values 960
 as filters 100
 GMT
 about 960
 converting 960
 official eBay time 76
 on eBay sites 960
 usage in schema 959
 UTC
 converting 960
title
 US eBay Motors listings 568
tokens
 defined 117
 expiration 128
 source examples 127
 using 101, 127
transactions
 defined 145
 Combined Payment 183
 retrieving 418, 419
troubleshooting
 getting help 35
types
 abstract 75
 concrete 76
 message 75

U

unanswered questions 404
units of measure, modeled 78
Unpaid Item Assistance 463
Unpaid Item Dispute
 diagram 460
Unpaid Item Process
 managing 455
unpaid item process
 AddDisputeResponse 775
 closing dispute 467
 DisputeID 466
 DisputeState 466
 getting information 469
 retrieving dispute 470
 reversing dispute 470
 SellerReverseDispute 922
updating
 WSDL 84, 86
UPI 460
UPI Assistance 463
UPI diagram 460
UploadSiteHostedPictures 941
URLs
 eBay Motors (US)
 eBay sites 151
 eBay Stores 154
 PayPal 153
 Production Gateway 87
 routing the request 87
 Sandbox Gateway 87

US eBay Motors 559
 deposit 568
 Immediate Payment 569
 limited warranty 570
 Manufacturer attribute 567
 Model attribute 567
 payment methods 568
 searching for listings on 571
 site ID 559
 subtitle 568
 title 568
use cases
 supported in this release 47
used item condition 218
user
 Seller Dashboard 520
user information, managing 513
username 62
users
 defined 145
 authentication by password 103
 authentication by token 101
 authentication, explained 117
 automated Sandbox test users 63
 bidders 148
 buyers 148
 creating a test user 63
 Half.com 575
 retrieving 513
 roles 147
 sellers 148
 Store owners 148
UTC
 converting to and from 960
UTF-8 character set 70

V

ValidateChallengeInput 942
ValidateTestUserRegistration 943
validation
 explained 80
 international differences 590
 string length restrictions 82
values
 predefined in code lists 80
 why some are ignored 80
variations 264
vehicles
 cross-promoting on mobile.de 572
VerifyAddFixedPriceItem 944
VerifyAddSecondChanceItem 946
VerifyRelistItem 947
VerReportItems
 defined 948
version
 compatibility levels 71
 gateway URL and 89
 release 84
 schema 95
VIN

for Australia 573

W

- Want It Now** 309
- WatchedItemEndingSoon**
 - using 635
- WatchedItemEndingSoon notification** 663, 763
- Web services**
 - defined 953
 - about eBay 45
 - invoking 74
- Wireless** 634
- wireless**
 - defined 634
- WSDL**
 - generating interfaces and stubs 51
 - location of eBay's 51
 - when to update 84, 86

X

- XML API**
 - defined 45
 - documentation, location of 45