ORACLE®

Advanced Commerce Concepts

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Oracle Training Materials – Usage Agreement

Use of this Site (“Site”) or Materials constitutes agreement with the following terms and conditions:

1. Oracle Corporation (“Oracle”) is pleased to allow its business partner (“Partner”) to download and copy the information, documents, and the online training courses (collectively, “Materials") found on this Site. The use of the Materials is restricted to the non-commercial, internal training of the Partner’s employees only. The Materials may not be used for training, promotion, or sales to customers or other partners or third parties.

2. All the Materials are trademarks of Oracle and are proprietary information of Oracle. Partner or other third party at no time has any right to resell, redistribute or create derivative works from the Materials.

3. Oracle disclaims any warranties or representations as to the accuracy or completeness of any Materials. Materials are provided "as is" without warranty of any kind, either express or implied, including without limitation warranties of merchantability, fitness for a particular purpose, and non-infringement.

4. Under no circumstances shall Oracle or the Oracle Authorized Delivery Partner be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this Site of Materials. As a condition of use of the Materials, Partner agrees to indemnify Oracle from and against any and all actions, claims, losses, damages, liabilities and expenses (including reasonable attorneys' fees) arising out of Partner’s use of the Materials.

5. Reference materials including but not limited to those identified in the Boot Camp manifest can not be redistributed in any format without Oracle written consent.

Agenda

- Abandoned Order Service
- B2B Features of ATG

Learning Objectives

At the end of this lesson you should be able to:

- Learn about the Abandoned Order Subsystems
- Understand the Abandoned Order Repository Extensions
- Learn about Abandonment scenarios
- Understand the B2B Features in ATG
- Learn about Invoices, Requisitions and Contracts

Section 1:
Abandoned Order Service

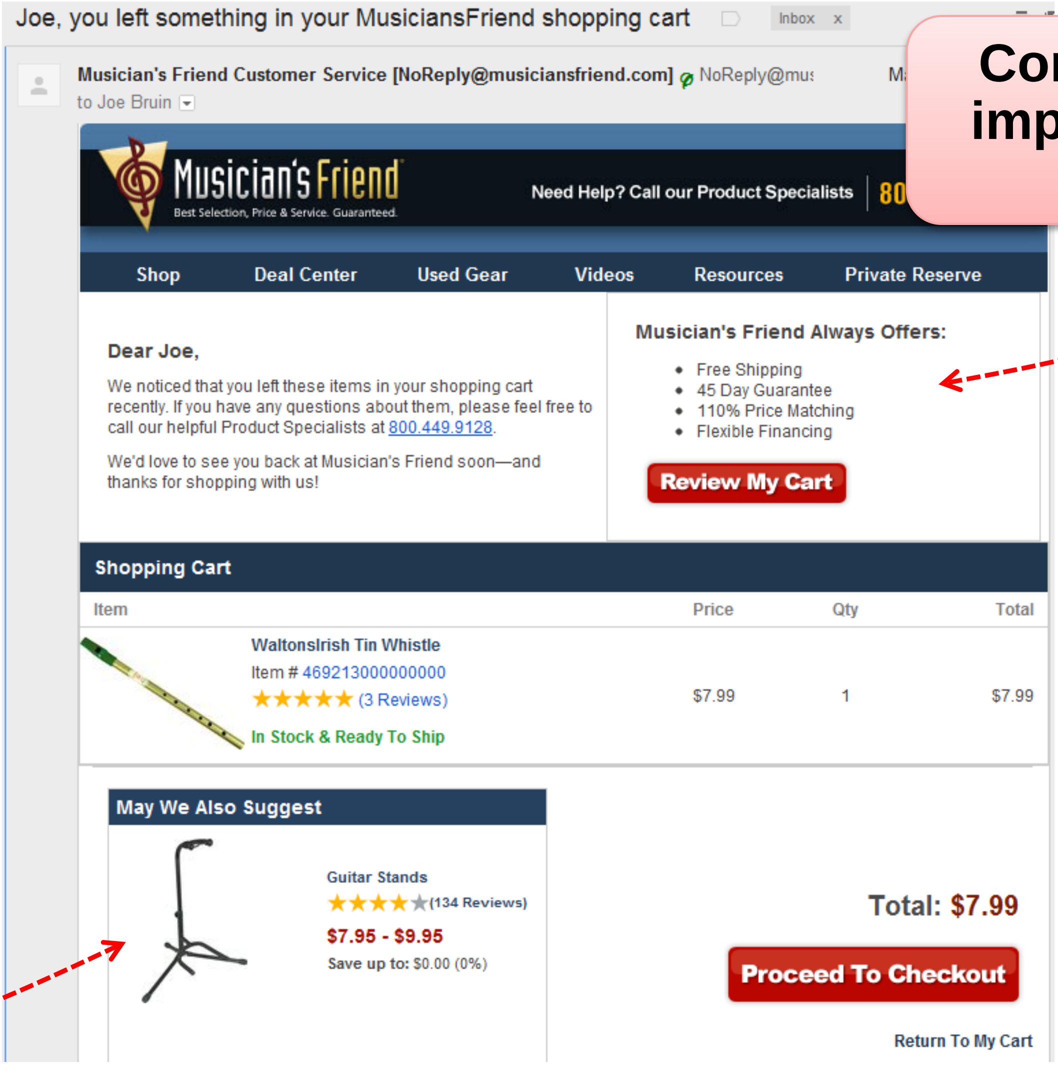
Abandoned Order Service

- An abandoned order or shopping cart is one that a customer creates and adds items, but never checks out.
- The Abandoned Order Service module is provided to:
  - Detect abandoned orders,
  - Respond to abandoned orders,
  - Report on order abandonment and related activities.
- Abandoned order module enables you to better understand what kind of orders your customers are abandoning.
- It allows you to campaign effectively to entice them to reclaim and complete them.
- It results in increased order conversion and revenue.

Example of Order Abandonment

- Joe Bruin visits an eCommerce site to buy a tin whistle.
- He proceeds to browse for it and places it in the cart.
- He then realizes that he could probably try to get it cheaper elsewhere. Besides, he is not sure if this is the right one for him.
- He navigates away and does not return for the next 7 days. The order is considered **abandoned**.
- The eCommerce site sends him an email after 7 days reminding him that he could still order the whistle. They may include a promotion such as free shipping.
- He returns to the site, **reanimating** the order.
- If he checks out, the order is considered **converted**.
- If he does not for 30 days, the order is considered **lost**.

Abondoned Order Email Sample



Joe, you left something in your MusiciansFriend shopping cart

Inbox x

Musician's Friend Customer Service [NoReply@musiciansfriend.com] NoReply@mus

to Joe Bruin

Musician's Friend

Best Selection, Price & Service. Guaranteed.

Need Help? Call our Product Specialists 80

Shop | Deal Center | Used Gear | Videos | Resources | Private Reserve

Dear Joe,

We noticed that you left these items in your shopping cart recently. If you have any questions about them, please feel free to call our helpful Product Specialists at 800.449.9128.

We'd love to see you back at Musician's Friend soon—and thanks for shopping with us!

Musician's Friend Always Offers:

- Free Shipping
- 45 Day Guarantee
- 110% Price Matching
- Flexible Financing

Review My Cart

Shopping Cart

Item	Price	Qty	Total
WaltonsIrish Tin Whistle Item # 469213000000000 ★★★★★ (3 Reviews) In Stock & Ready To Ship	$7.99	1	$7.99

May We Also Suggest

Guitar Stands

★★★★★(134 Reviews)

$7.95 - $9.95

Save up to: $0.00 (0%)

Total: $7.99

Proceed To Checkout

Return To My Cart

Conversion improvement text

Abondoned Item

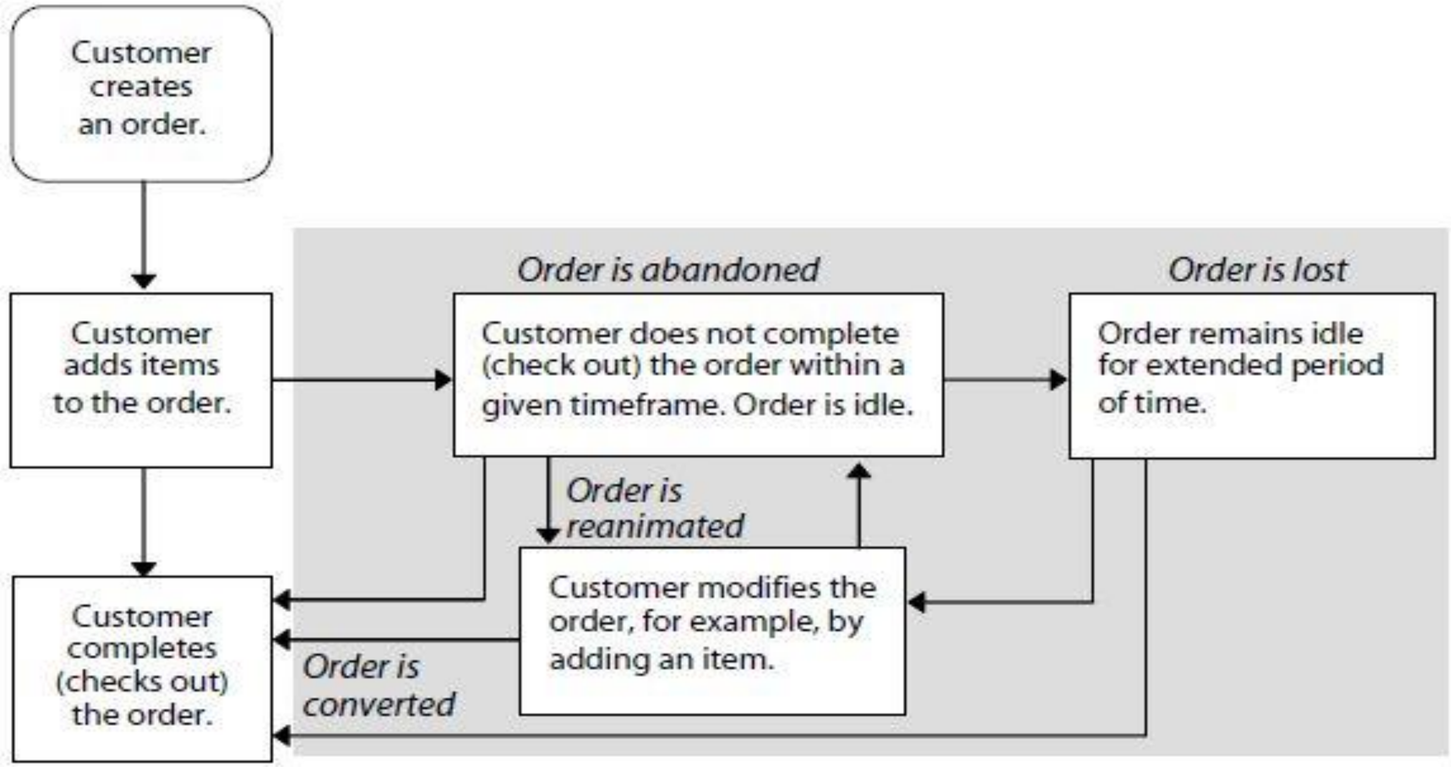
Cross Sell

For Oracle employees and authorized partners only. Do not distribute to third parties.

How Does It Work

- When an item is placed in a cart, the **orderLastUpdated** property of the **abandonmentInfo** item of order is updated with the date the order was modified by the user.
- A scheduler service called **AbandonedOrderService** runs nightly, checking to see if this date is a configured number of days in the past. It then marks the order as abandoned and fires the **order abandoned event**.
- The scenario engine receives this event and sends an email to the customer.
- When the customer reanimates and converts the order, another event (**Abandoned Order is Converted**) is fired.
- The scenario engine marks the order as converted and logs the conversion to the **AbandonedOrderLog repository.**

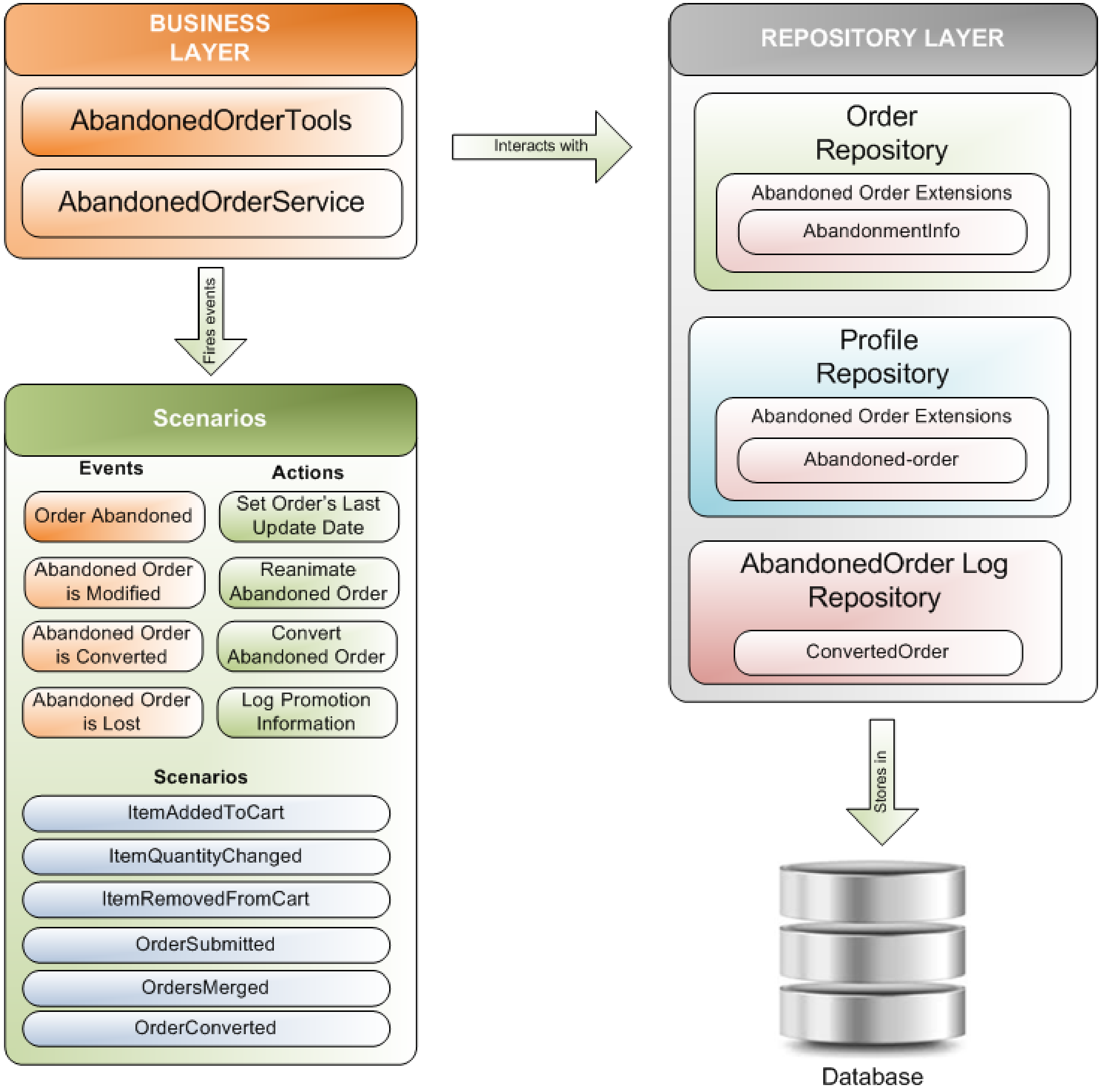
Overview of Abandoned Order


Customer creates an order.
Customer adds items to the order.
Order is abandoned
Customer does not complete (check out) the order within a given timeframe. Order is idle.
Order is lost
Order remains idle for extended period of time.
Order is reanimated
Customer modifies the order, for example, by adding an item.
Order is converted
Customer completes (checks out) the order.

Abandoned Orders States

- **Abandoned orders**: Incomplete orders that have not been checked out (ABANDONED).
- **Reanimated orders**: Previously abandoned orders that have since been modified by the customer (REANIMATED).
- **Converted orders**: Previously abandoned orders that have been successfully checked out (CONVERTED).
- **Lost orders**: Abandoned orders that have been abandoned for so long that reanimation is no longer realistic (LOST).

Abandoned Order Subsystems

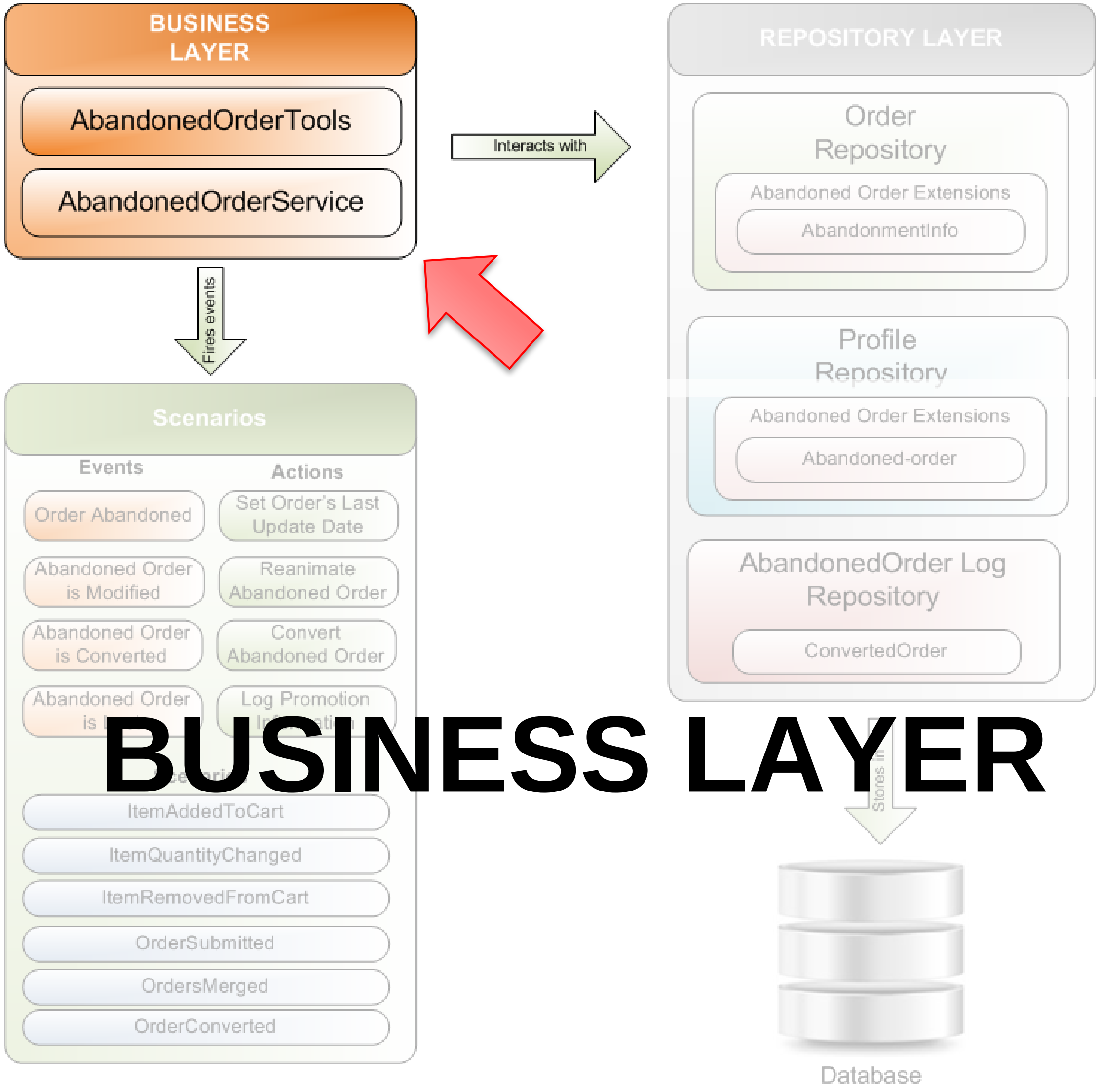


Abandoned Order Subsystems

- The core of the Abandoned Order System is the business layer. This consists of a scheduleable service called AbandonedOrderService and a tools class.
- It interacts with the order repository, profile repository, and the AbandonedOrder log repository.
- The order repository and the profile repository have Abandoned Order Extensions to track the status.
- The business layer fires events that are handled by the scenario engine. There are scenario event and action extensions specific to the Abandoned Order System.

For Oracle employees and authorized partners only. Do not distribute to third parties.

BUSINESS LAYER



For Oracle employees and authorized partners only. Do not distribute to third parties.

AbandonedOrderService Component

- AbandondedOrderService is a scheduled service with two important functions:
  - Storing the criteria to identify abandoned and lost orders.
  - Querying the order repository to identify orders as abandoned or lost.
- The Service can be configured at /atg/commerce/order/abandoned/AbandonedOrderService.
- Important properties of the service are:
  - idleDaysUntilAbandoned and idleDaysUntilLost,
  - minimumAmount,
  - schedule.

Defining Abandoned and Lost Orders

- You set these criteria for abandoned and lost orders in the following properties of the AbandonedOrderService component:
  - **idleDaysUntilAbandoned**: Number of days before an incomplete order becomes abandoned.
  - **idleDaysUntilLost**: Number of days before an incomplete order becomes lost.
  - **minimumAmount**: The minimum amount in consideration for abandoned and lost orders. This field is optional.
- You can extend these components and override OOTB methods to provide more functionality.

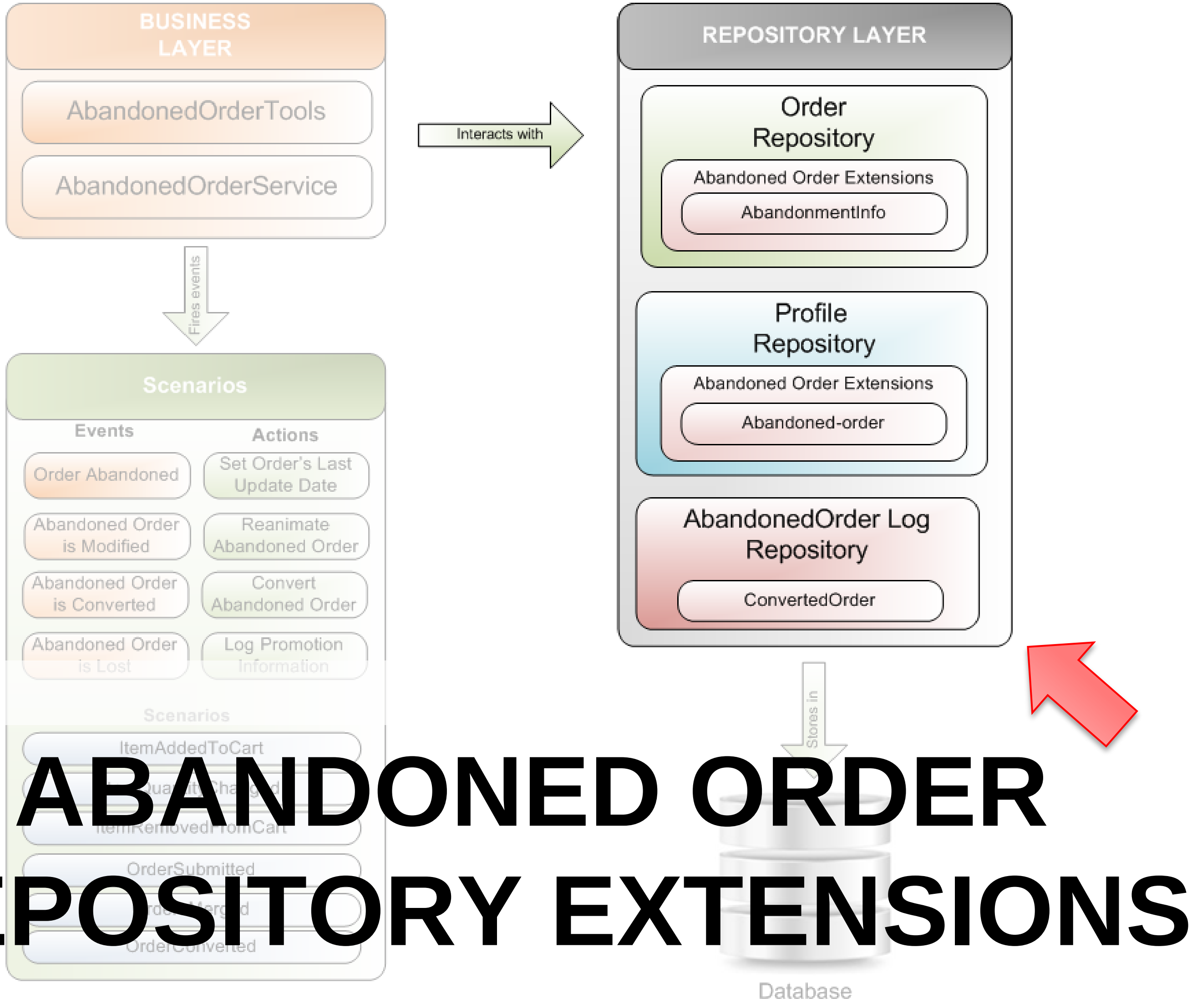
Detecting Abandoned Orders

- AbandonedOrderService detects both abandoned and lost orders:

Criteria for ABANDONED	Criteria for LOST
Order state is **INCOMPLETE.**	Order state is **INCOMPLETE.**
Abandonment state is **REANIMATED.**	Abandonment state is **not LOST.**
The order has been idle for at least **idleDaysUntilAbandoned.**	The order has been idle for at least **idleDaysUntilLost.**
The order’s subtotal is greater than or equal to **minimumAmount.**	The order’s subtotal is greater than or equal to **minimumAmount.**

AbandonedOrderTools Component

- AbandonedOrderTools component stores the central configuration for the entire Abandoned Order Service module.
- It can be configured at /atg/commerce/order/abandoned/AbandonedOrderTools.
- Important properties of the component groups are:
  - Properties that store state names,
  - Properties that store item names,
  - Properties that store property names,
  - Message related properties,
  - Other import properties.



ABANDONED ORDER REPOSITORY EXTENSIONS

Abandoned Repository Extensions to Order Repository

- The Abandoned Order Services module extends the repository definition for the order repository by adding:
  - An additional property named abandonmentInfo to the order item descriptor which stores an item of type abandonmentInfo.
  - An additional item descriptor named **abandonmentInfo** which stores the abandonment information for an order. The following table describes each abandonmentInfo property:
    - order,
    - orderLastUpdated,
    - state,
    - abandonmentDate/ conversionDate/ reanimationDate/ lostDate,
    - abandonmentCount.

Abandoned Order Extensions to the Profile Repository

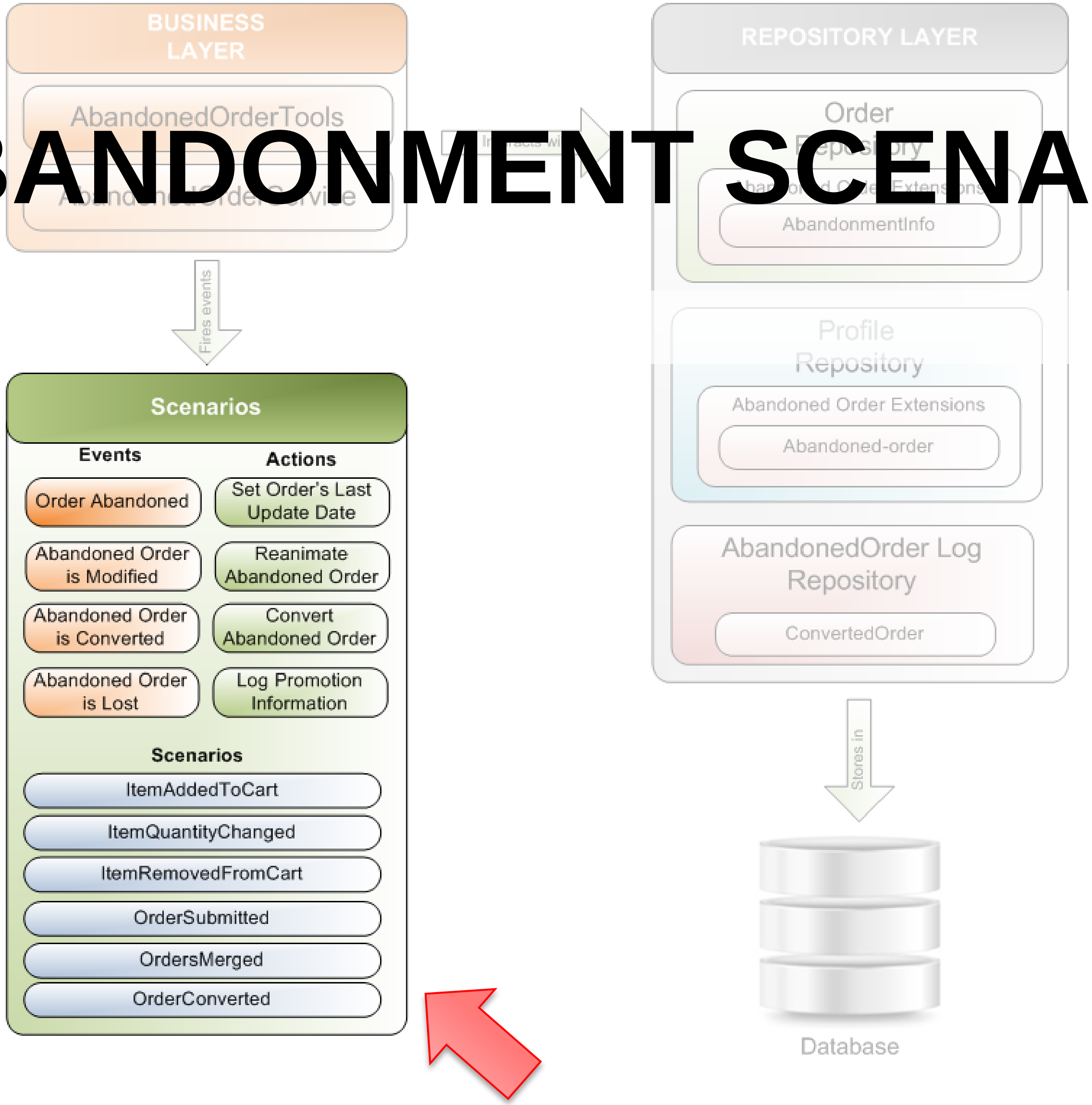
- The Abandoned Order Services module extends the repository definition for the profile repository by adding:
  - An additional item descriptor named abandoned-order. Items of this type have two properties:
    - orderId, which stores the ID of the abandoned order.
    - profileId, which stores the ID of the user profile associated with the abandoned order.
  - Two additional properties to the user item descriptor:
    - abandonedOrders, which stores the list of abandoned-order items currently associated with the user.
    - abandonedOrderCount, which is a derived property that stores the number of items in the abandonedOrders user property.

The Abandoned Order Log Repository

- Abandoned order log repository stores information about converted orders.
- It defines a single item descriptor named ConvertedOrder.
- The following are the properties of the ConvertedOrder:

Property	Description
orderId	The ID of the converted order.
convertedDate	The date and time that the order was converted.
amount	The total price of the converted order.
promotionCount	The number of promotions that were applied to the converted order.
promotionValue	The total value of the promotions that were applied to the converted order.

ABANDONMENT SCENARIOS



BUSINESS LAYER

AbandonedOrderTools

AbandonedOrderService

Interacts with

Fires events

REPOSITORY LAYER

Order Repository

Abandoned Order Extensions

AbandonmentInfo

Profile Repository

Abandoned Order Extensions

Abandoned-order

AbandonedOrder Log Repository

ConvertedOrder

Stores in

Database

Scenarios

Events	Actions
Order Abandoned	Set Order's Last Update Date
Abandoned Order is Modified	Reanimate Abandoned Order
Abandoned Order is Converted	Convert Abandoned Order
Abandoned Order is Lost	Log Promotion Information

Scenarios

- ItemAddedToCart
- ItemQuantityChanged
- ItemRemovedFromCart
- OrderSubmitted
- OrdersMerged
- OrderConverted

For Oracle employees and authorized partners only. Do not distribute to third parties.

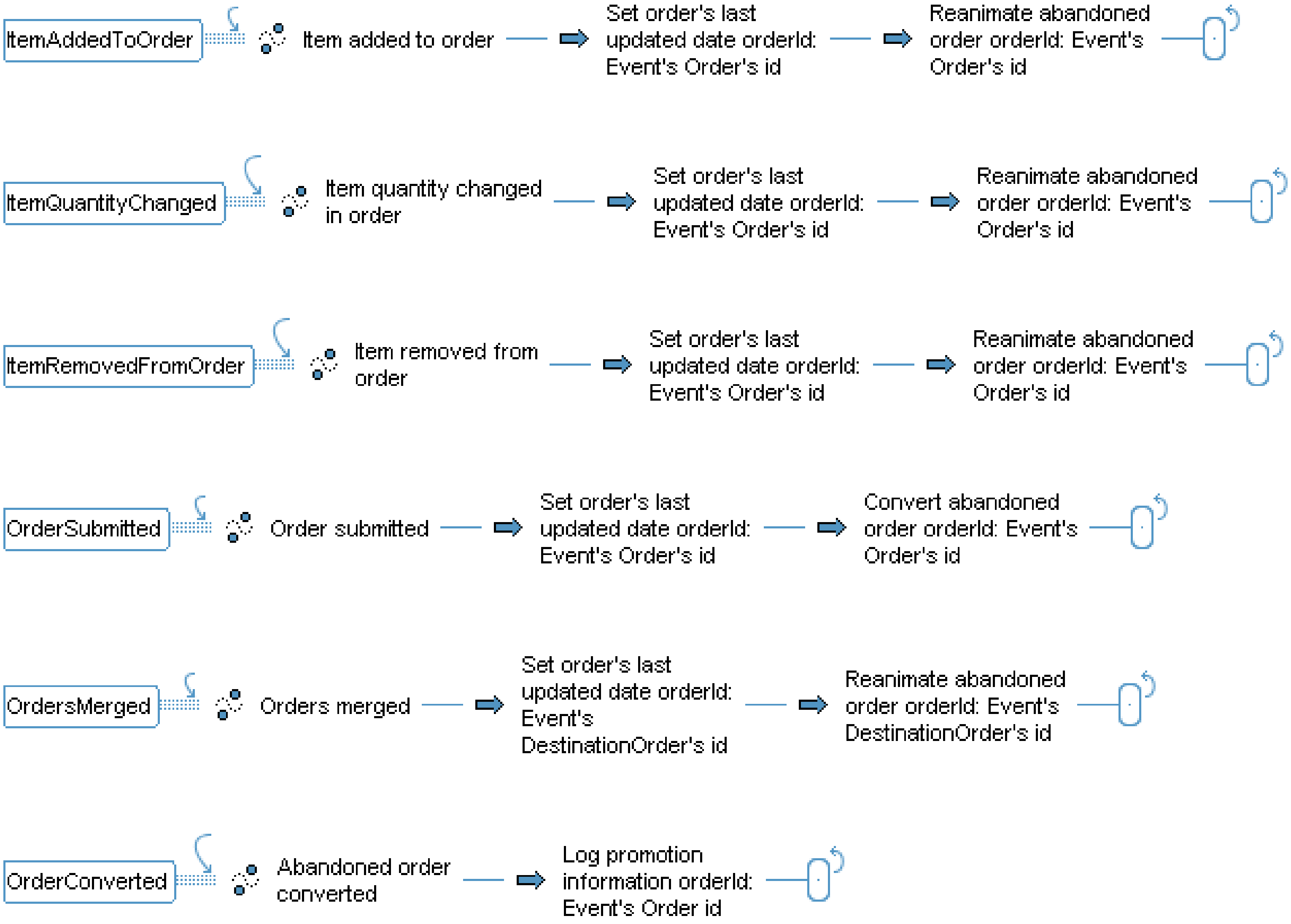
Abandonment Scenario Events

- The Abandoned Order Services module includes the following scenario events:
  - **Order Abandoned**: Triggered when an order is identified as abandoned.
  - **Abandoned Order is Modified**: Triggered when an abandoned order is modified by owner.
  - **Abandoned Order is Converted**: Triggered when an abandoned order is checked out.
  - **Abandoned Order is Lost**: Triggered when an abandoned order is identified as lost.

Abandonment Scenario Actions

- The Abandoned Order Services module includes the following scenario actions to respond to user activity:
  - **Set Order's Last Updated Date**: Updates the orderLastUpdated property of the AbandonedInfo item of the order.
  - **Reanimate Abandoned Order**: Reanimates the order and fires relevant events.
  - **Convert Abandoned Order**: Converts the abandoned order and fires relevant events.
  - **Log Promotion Information**: Logs the promotion related information for a converted order.

Abandonment Scenarios (1)



ItemAddedToOrder → Item added to order → Set order's last updated date orderId: Event's Order's id → Reanimate abandoned order orderId: Event's Order's id

ItemQuantityChanged → Item quantity changed in order → Set order's last updated date orderId: Event's Order's id → Reanimate abandoned order orderId: Event's Order's id

ItemRemovedFromOrder → Item removed from order → Set order's last updated date orderId: Event's Order's id → Reanimate abandoned order orderId: Event's Order's id

OrderSubmitted → Order submitted → Set order's last updated date orderId: Event's Order's id → Convert abandoned order orderId: Event's Order's id

OrdersMerged → Orders merged → Set order's last updated date orderId: Event's DestinationOrder's id → Reanimate abandoned order orderId: Event's DestinationOrder's id

OrderConverted → Abandoned order converted → Log promotion information orderId: Event's Order id

Abandonment Scenarios (2)

- The out-of-the-box scenarios simply maintain state of the order and collect logs.
- These can be changed to communicate with the customer or give them a promotion.
- New ones can be created using the events and actions provided by the Abandoned Order Service.

Additional Considerations

- Abandoned Order Services module can track orders abandoned by transient users (anonymous users).
  - This can be helpful for analytics.
  - Anonymous user abandonment tracking can be turned off.
- Abandoned Order Services module can be extended.
  - For example, you can track high value abandonment from low value abandonment and take different actions for both.
  - You can modify the specific set of criteria used to identify the abandoned order beyond the basic properties provided.

Section 1 ✔
Check Your Understanding

What are the abandoned order states?

Answer:

ABANDONED, REANIMATED, CONVERTED, and LOST.

Section 1 ✓
Check Your Understanding

What does the AbandonedOrderService component do?

Answer:

It is a scheduled service that queries the repository for abandoned orders.

Section 1 ✔
Check Your Understanding

When does an order become LOST?

Answer:

After a configurable amount of time has passed since the last update to the order by the user.

Section 1 ✔
Check Your Understanding

What repositories are extended by the Abandoned Order Service?

Answer:

The profile and order repositories. In addition an AbandonedOrderLogRepository is added.

Section 1 ✔
Check Your Understanding

Name a few abandonment scenario events.

Answer:

Order Abandoned, Abandoned Order is Modified, Abandoned Order is Converted, Abandoned Order is Lost.

Summary

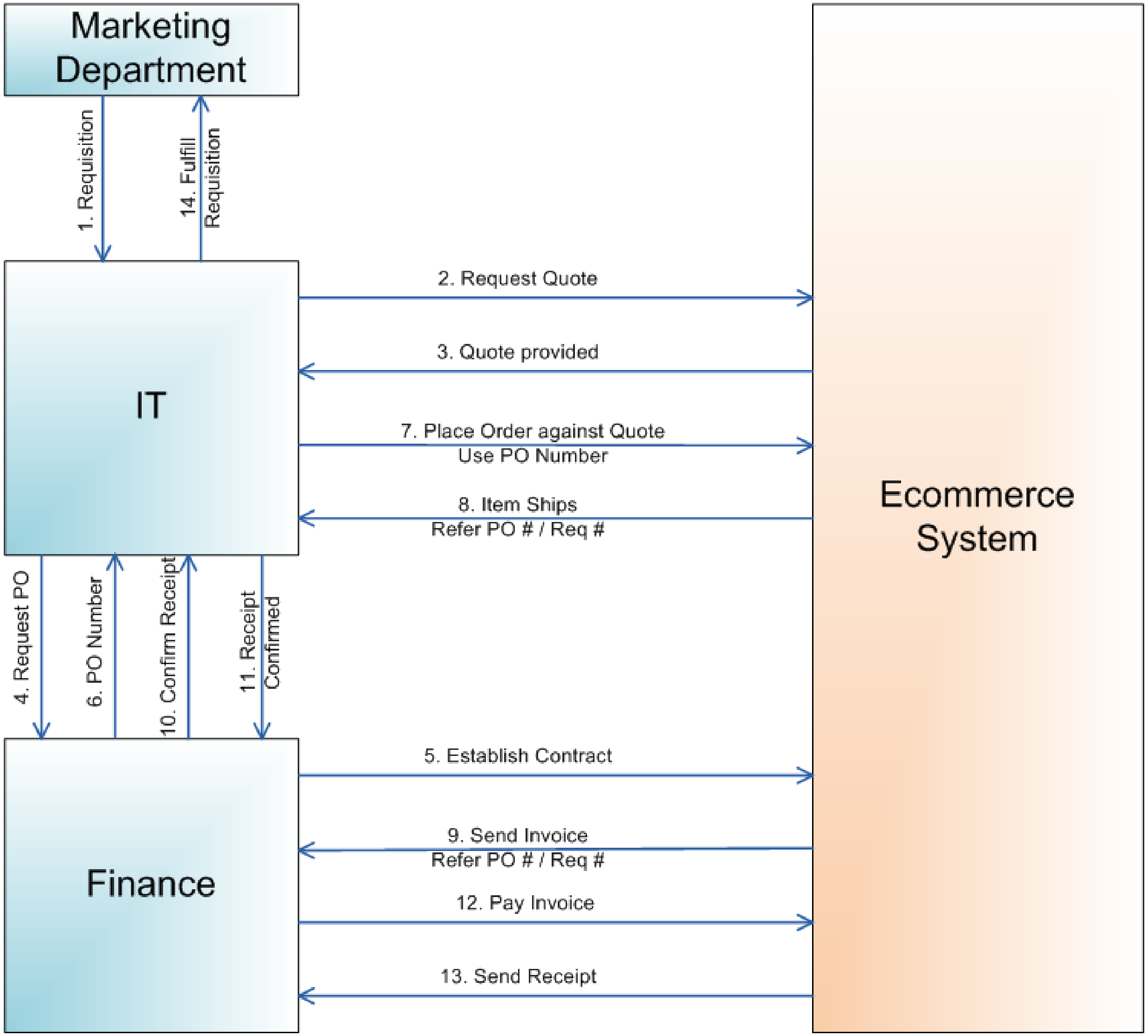
- An abandoned order or shopping cart is one that customer creates and adds items, but never checks out.
- You can use the Abandoned Order Service to detect, respond, and report on these orders.
- Abandoned Order Service consists of business layer, repository, and scenario subsystems.
- AbandonedOrderService and AbandonedOrderTools components are present in the business layer.
- Abandoned Order System has extensions to order and profile repository and AbandonedOrderLogRepository.
- It exposes events and actions to allow the application to respond and take appropriate action.

Section 2:
B2B Features of ATG

Understanding B2B Processes

- B2B refers to Business-to-Business. It is used to represent commerce between two businesses as opposed to B2C which is between a business and a consumer.
- A lot of B2B processes occur offline and are of wide variety.
- When dealing with B2B, a typical process is to ship the goods before payment is made. This depends on the contract between the businesses.
- ATG supports a limited set of features to facilitate B2B Commerce.

Example B2B Business Process (1)



For Oracle employees and authorized partners only. Do not distribute to third parties.

Example B2B Business Process (2)

1. An employee in a company requires a personal computer. He makes a requisition to the IT department typically by filling out a requisition form.
2. IT assigns a **requisition id** to the request. Multiple requests may be grouped under this requisition spanning multiple orders. They request a quote from the eCommerce System.
3. eCommerce System presents a **quote**. This is a guarantee to supply the merchandise at a quoted price within a specified time.
4. The IT department turns the quote over to Finance department and requests a **Purchase Order (PO).**
5. The Finance department may establish a **contract** with the eCommerce system, if one is not already in place.

Example B2B Business Process (3)

6. The Finance department then generates a **Purchase Order (PO)** number for the requested amount and forwards it to IT.
7. IT turns the quote into an **order**. They reference the PO number and/ or requisition ID in their order.
8. The eCommerce System ships the item against the PO which is treated as a promise to pay.
9. They then send an **invoice** to the Finance Department.
10. and 11. The Finance department request and receives confirmation that the shipment is acceptable.
12. and 13. The invoice is paid and a receipt is received.
14. The shipment which contains the personal computer is sent to the marketing department at some point.

B2B Commerce Features

- Typically, a lot of the described process is manual and happens outside the system.
- Also, there is a large amount of variance from company to company.
- ATG B2B Commerce provides the following features:
  - **Invoice**: Allows for generation of an invoice using purchase order number and billing address.
  - **Requisition**: Another way for a customer to track a purchase through the website.
  - **Contracts**: Allows for associating a particular catalog, price lists, and payment terms with a specific organization.
- Quote generation or handling is **not** supported in ATG.
- Invariably, developers implementing B2B need to customize the framework to meet their business requirements.

Invoice Overview

- The steps involved in invoice creation and processing are:
  - A user asks for an invoice to be generated upon checkout and provides a purchase order number. An invoiceRequest payment group is created and attached to the order.
  - When the order is fulfilled, the PaymentManager performs a debit by calling on the InvoiceManager to generate a new invoice object.
  - The invoice is created and stored in the invoice repository.
  - JMS messages are generated informing the Dynamo Scenario Manager and any other modules you designate that an invoice has been created.
- Developers must leverage this message to inform relevant parties that an invoice is available.

Invoices in Checkout

- The InvoiceRequest object represents the customer’s request to be billed for a purchase.
  - It implements the PaymentGroup interface and extends PaymentGroupImpl.
  - Its properties: purchase order number, billing address, invoice format, delivery mode, payment due date, and payment terms.
- Use the atg.b2bcommerce.order.purchase. **CreateInvoiceRequestFormHandler** class to create an InvoiceRequest payment group.
- Invoice validation is done as part of the ValidateForCheckout pipeline chain.

Invoice Payment

- The InvoiceRequestProcessor does the work of creating invoices.
- If you want to add further validation logic to your invoice processing, you should extend the InvoiceRequestProcessor.authorize() method.
- The InvoiceManager class provides high-level access for creating, manipulating, saving, and deleting invoice objects in the invoice repository.
- The InvoiceManager class executes pipelines whenever an invoice is created, loaded, updated, or removed.
  - **Pipeline Chains:** addInvoice, updateInvoice, removeInvoice.

The Invoice Repository

- The invoice repository uses SQL repository which appears at /atg/commerce/invoice/InvoiceRepository.
- Its configuration file is located at <ATG10dir>/DCS/src/config/atg/commerce/invoice/invoicerepository.xml.
- Repository items in invoice repository:
  - **Invoice**: Stores base info of invoice including ID, creationDate, balanceDue, invoiceNumber, requisitionNumber, and so on.
  - **DeliveryInfo**: Descriptor represents physical or electronic delivery information.
  - **PaymentTerms**: Represents payment terms for an order, expressed in terms of discount percentage, discount days, and net days.

Requisitions Overview

- Requisition numbers provide another way for buying organizations to track orders from the purchase process through billing.
- ATG Business Commerce keeps track of the requisition number and carries it over to the billing process when invoices are generated.
- ATG Business Commerce classes include features to permit use of requisitions:
  - The PaymentGroup object includes the **requisitionNumber** property.
  - Multiple payment groups may use the same requisitionNumber.

Contracts Overview

- Contracts allow you to associate the following to a specific organization:
  - Catalog,
  - Price lists,
  - Payment terms.
- Invoices in ATG Business Commerce are automatically set up to handle contracts.
- When a new InvoiceRequest is created, the following is copied over from the contract in the user organization's information:
  - Payment Net Days,
  - Payment Discount Days,
  - Payment Discount Percent.

Contracts Repository Items

- The **contract** repository item has the following important properties:
  - catalog: Catalog associated with the contract.
  - endDate: Date when the contract ends.
  - negotiatorInfo: ContactInfo item for the negotiator.
  - priceList: Applicable pricelist associated with the contract.
  - terms: A contract terms repository object.
- The **ContractTerms** repository item has:
  - paymentDiscountPercent: Discount if the invoice is paid within discountDays period.
  - paymentDiscountDays: Days within which the discountPercentage applies.
  - paymentNetDays: Time at which payment in full of the net price is due.

Section 2 ✓
Check Your Understanding

What are the three B2B features currently supported in ATG B2B Commerce?

Answer:

Invoice, Requisition, and Contracts.

Section 2
Check Your Understanding

What is the InvoiceRequest object?

Answer:

The InvoiceRequest object represents the customer’s request to be billed for a purchase. It implements the PaymentGroup.

Section 2 ✓
Check Your Understanding

Which method can extended to add further validation logic to your invoice processing?

Answer:

InvoiceRequestProcessor.authorize() can be extended to add more logic.

Section 2 ✓
Check Your Understanding

Where is the requisitionNumber stored on the commerce objects?

Answer:

It is stored on the PaymentGroup object.

Section 2 ✔
Check Your Understanding

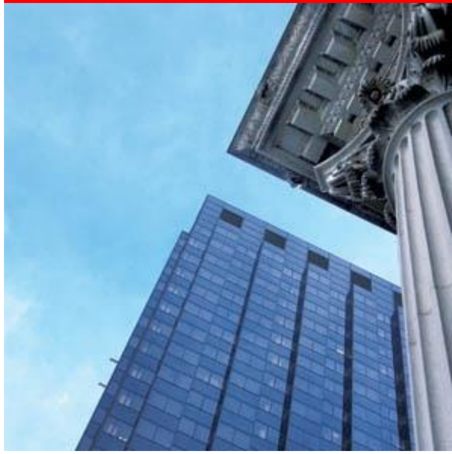
What are the repository items provided in the contract repository?

Answer:

Contract and ContractTerms repository items.

Summary

- B2BCommerce used to represent commerce between two businesses.
- Invoice allows for generation of an invoice using purchase order number and billing address.
- Requisition is another way for a customer to track a purchase through the website.
- Contracts allows for associating a particular catalog, price lists, and payment terms with a specific organization.



Q&A

For Oracle employees and authorized partners only. Do not distribute to third parties.

ORACLE IS THE INFORMATION COMPANY

ORACLE®