



Quick Start Guide

Wild West Domains® API Reseller Quick Start Guide Web Service Edition

Wild West Domains API Reseller Quick Start Guide, Web Service Edition (Version 1.1) Copyright © 2009. All rights reserved.

Distribution of this work or derivative of this work is prohibited unless prior written permission is obtained from the copyright holder. All trademarks and copyrights of products discussed in this document are the property of their respective owners.

Intellectual Property. You agree that Wild West Domains and its parent and affiliate companies own all proprietary rights, including but not limited to copyrights, patents and trade secrets, trademarks, and service marks, in and to the API and that this Agreement does not transfer ownership of any of these rights. Wild West Domains expressly reserves its rights in and to all such content and materials. Wild West Domains shall own all proprietary rights in any modifications to the API, whether created by Wild West Domains, users, or a third party. You hereby assign to Wild West Domains all proprietary rights, including copyright, patent and trade secret rights, to any modifications created by You. No license or right under any copyright, patent, trademark, service mark or other proprietary right or license is granted to You or conferred upon You by this Agreement or otherwise. You will use all computer programs, documentation and information consisting of or containing proprietary information related to the API solely for the purpose of performing under this Agreement. You will not decompile, disassemble or otherwise reverse engineer the API or cause others to do so. You will not modify or cause others to modify the API, without the prior written consent of Wild West Domains. Except as otherwise explicitly agreed in writing, Wild West Domains-owned content described in this document may be downloaded, displayed, reformatted and printed for Your personal, non-commercial use only. You agree to prevent any unauthorized copying of the API. Content owned by Wild West Domains advertisers, suppliers or licensors may be subject to additional restrictions. You agree to use the API and this document only within the scope of their intended business purposes.

Limit on Wild West's Obligation. Provision of any API under this Agreement shall not create any obligation for Wild West to continue to develop, support, repair, offer for sale or in any other way continue to provide or develop the API. Furthermore, the API may contain defects. By using this Software, you agree that you are aware that it may contain defects. You are advised to safeguard Your important data, to use caution and to not rely in any way on the correct functioning or performance of the API.

Warranties. Wild West Domains provides the API "as is" and without warranty or guarantee of any kind, either express or implied, including but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose. In no event shall Wild West Domains be liable for any loss of profits, loss of business, loss of data, unsecured transactions, interruption of business, or for indirect, special, or consequential damages of any kind, even if Wild West Domains has been advised of the possibility of such damages arising from any defect or error in the API. In no way event shall Wild West's liability exceed the initial license fee for the API paid by You.

Indemnification. You shall indemnify, defend by counsel reasonably acceptable to Wild West Domains, protect and hold Wild West Domains harmless from and against any and all claims, liabilities, losses, costs, damages, expense, including consultants' and attorneys' fees and court costs, demands, causes of action, or judgments directly or indirectly arising out of or related to Your use of the API.

1	Introduction	1
	Quick Start Icons	1
	Web Service API	2
	Web Service API Advantages	2
	Code Snippet	3
	System Requirements	3
	Technical Restrictions	3
	Other Restrictions	3
	Technical Support	3
	Quick Start Definitions	4
	Operational Testing Environment	4
	Integration Checklist	4
	Tutorials	4
	Certification Primers	4
2	Integration Checklist	5
	Pre-Certification (OTE)	5
	Designate IP Addresses	5
	Connect to the WWD API Server	5
	Certification (OTE)	5
	Register the reseller account for certification	6
	Complete the seven certification tasks	6
	Code Snippet	7
	Domain Name Availability Check (Task #1)	7
	Domain Name Registration (Task #2)	7
	Additional Information:	7
	Other Data	8
	Domain Name Privacy Purchase (Task #3)	8
	Additional Information:	9
	User	9
	Domain resource ID	9
	Domain Name Availability Check (Task #4)	9
	Domain Name Information Query (Task #5)	9
	Domain Name Renewal (Task #6)	10
	Additional Information:	10
	Domain Name Transfer (Task #7)	10
	Additional Information:	10
	Pass Certification	11
	Post-Certification (Live Environment)	11
	Update the Web Reference	11
	Update the Reseller Extranet	12
	Wire account funds	12
	Web Service Method Format	12

ciTRID (Client Transaction Identifier)	13
Cred (Credential Object)	13
Return XML Format	13
Response XML Format (Request Success)	13
Response XML Format (Request Failure)	14
3 Tutorials and Code Samples	15
Visual Studio .NET	15
To add a Web reference to the Web service API:	15
C# Code Sample: Invoking the Describe Method	16
VB.NET Code Sample: Invoking the Describe Method	17
PHP Code Sample: Invoking the Describe Method	18
Java Code Sample: Invoking the Describe Method	19
4 Certification Primers	20
Certification Primer - Task #1 and #4: CheckAvailability	20
Certification Primer - Task #2: Domain Name Order Request	22
Certification Primer - Task #3: Privacy Order Request	30
Certification Primer - Task #5: Domain Info Request	33
Certification Primer - Task #6: Domain Renewal Order Request	36
Certification Primer - Task #7: Domain Transfer Order Request	38
5 Poll Requests	40
6 API Return Code Values	43
7 Web Service Methods Overview	45

1. Introduction

The goal of the Wild West Domains (WWD) API Reseller Quick Start Guide is to help resellers integrate their Web sites and applications with the Web Service API as quickly and easily as possible. Areas covered include: code samples to reference the Web service, passing certification, and transitioning to the production API environment.

The Quick Start Guide is divided into three sections:

- **Integration Checklist** - Provides the mission-critical path for getting the API successfully integrated with an existing reseller sales Web site.
- **Tutorials** - Offer detailed explanations of particular API topics
- **Certification Primers** - Deliver concrete instructions on successful completion of specific certification tasks.

Quick Start Icons



Details process that must be implemented as described.



Outlines best practices and coding standards.



Offers additional pertinent information

WEB SERVICE API

This method of connection simplifies the overall development experience enabling resellers to be up and running in a fraction of the time otherwise required to developing a reseller Web site or application using SSL socket communications.

Web Service API Advantages

Simplified programming interface. The Web service exposes methods instead of assembling input XML. It also exposes types (classes or objects) or hierarchies of types (that holds transaction information and are used as parameters).

Note: The method return values are still XML code that needs to be parsed. Identical to the response returned when using an SSL socket connection.

Flexible programming interface. Resellers who prefer to work with XML to package transaction request information can use the ProcessRequest() method.

The ProcessRequest() method accepts the same XML format (used in a secure socket connection) except that the root node <wapi> requires two extra attributes: account and pwd. The account attribute holds the WWD account ID; and the pwd attribute holds the WWD password.

Note: The wapi/login xml element can no longer be used in ProcessRequest() as it is already passing in the account information for each transaction call.

Code Snippet

```
...  
  
WsWWDAPI.WAPI wapi = new WsWWDAPI.WAPI();  
  
sReturnXml = wapi.ProcessRequest("<wapi  
clTRID='transaction-id' account='MarkAPI'  
pwd='password'><describe/></wapi>");  
  
//Parse the return XML and display/process information  
  
...
```

System Requirements

To utilize the WWD Web Service API, competency in the following areas must be exhibited:

- Web site programming
- Shopping cart solution use
- Invocation of Web service methods (using SOAP)
- Parsing return XML strings
- Database (optional but highly recommended to store reseller transactions, user and product information)

Technical Restrictions

Connecting to WWD's API server using the Web service does not require the use of any specific programming language (e.g., C#, VB.NET, Java, PHP, etc.) as long as it supports the SOAP Web service protocol.

Other Restrictions

The only non-technical restrictions are the ICANN rules and other legalities agreed to in the reseller agreement.

Technical Support

WWD technical support is available to assist with API questions by email at APISupport@wildwestdomains.com.

QUICK START DEFINITIONS

Operational Testing Environment

The operational testing environment (OTE) is maintained by WWD to provide resellers a non-production environment in which to both refine their integration code and pass certification. Prior to passing certification, all reseller account management activity takes place in the OTE reseller extranet (<http://www.ote.resellerextranet.com>).

After certification, the reseller is granted access to the production API system, including the production extranet (<http://www.resellerextranet.com>).

Integration Checklist

The Integration Checklist outlines the necessary steps for connecting to and certifying through the API operational testing environment.

Tutorials

Tutorials provide sample codes and instructions on how to reference the WWD Web service API within mainstream programming environments (C#, VB.NET, PHP).

Certification Primers

Certification Primers exist to assist resellers through the certification process. They explain specific reseller requests required for the completion of specific certification tasks, including a detailed examination of the method parameters that are used.

2. Integration Checklist

PRE-CERTIFICATION (OTE)

These steps guide a reseller through successfully connecting to the API operational environment.

Designate IP Addresses

1. Navigate to the reseller extranet IP Address Configuration page (<http://www.ote.resellerextranet.com/account/IPAddresses.aspx>) using the account login information received in the WWD confirmation email with subject line "Your OTE account has been set up!"
2. Specify at least one secure IP address through which API connections will occur. (If the server IP address is unknown, open a command window and issue an ipconfig command to return the desired data.)
3. DO NOT supply a corresponding certificate name.



IGNORE THIS ENTRY AND KEEP IT BLANK. This is an optional feature that is for resellers using SSL socket connections. The Web service API does not need to configure SSL certificate to connect.

Connect to the WWD API Server

1. In your application project, add a Web Reference URL that points to OTE's WSDL (<https://api.ote.wildwestdomains.com/wswwdapi/wapi.asmx?wsdl>).
2. Once a Web reference is set, use the Describe(...) method to quickly test if the application can connect to the WWD server.

CERTIFICATION (OTE)

All resellers must demonstrate their competence with a series of API core requests before they are granted permission to connect to the production API environment. Upon the

completion of the API certification test, resellers receive a confirmation email message from WWD with further instructions.

Register the reseller account for certification

Navigate to the extranet Certification Status page (<http://www.resellerextranet.com/Account/CertificationStatus.aspx>). Check the box at the bottom of the page; then click Apply to enter into certification mode. With this setting in place, the API can evaluate reseller requests against its certification scripts.

Complete the seven certification tasks

The tasks described in sections 4.1.1-4.1.7 are completed in the OTE. See below for an overview of the seven tasks:

- Domain name availability check
- Domain name registration
- Domain name privacy purchase
- Domain name availability check
- Domain name information query
- Domain name renewal
- Domain name transfer



Certification tasks must be completed in their prescribed order as the API tracks the reseller's progression in the sequence. Failure to do so results in the error message "Request node(s) not matched."

Some tasks require information that can only be queried after the completion of previous tasks. For this reason it is important to keep a record of all API response XML. This can be accomplished automatically.



Once a task is successfully completed, it may not be repeated.

If the test needs to be reset in order to start over, call the `ProcessRequest(...)` method and use the script element of the manage request to reset.

Code Snippet

```
String sReturnXml, sCLTRID;

sCLTRID = System.Guid.NewGuid().ToString();
WsWWDAPI.WAPI wapi = new WsWWDAPI.WAPI();

sReturnXml = wapi.ProcessRequest("<wapi cLTRID=' " +
sCLTRID + "' account=' MarkAPI '
pwd=' password' ><manage><script cmd=' reset' /></manage></
wapi >");
```

Domain Name Availability Check (Task #1)

Instructions: Use the CheckAvailability() method to verify the availability of the domain names "example.us" and "example.biz".

Result: The API response XML should know that both domain names are available.



The domains must be listed in this order in the check request XML. The exact ordering of XML elements is only required during certification.

Domain Name Registration (Task #2)

Instructions: Use the OrderDomains() method to register the domain names "example.us" and "example.biz" for two years under a new shopper account, with password "abcde."

Result: The API response XML should return a message indicating that the two domains were successfully processed.

Additional Information:

Product IDs

- 2-year .us domain name registration - 350127
- 2-year .biz domain name registration - 350077

Name Servers

- ns1.example.com
- ns2.example.com

Registrant Data

- First name - Artemus
- Last name - Gordon
- Email - agordon@wildwestdomains.com
- Address - 2 N. Main St.
- City - Valdosta
- State - Georgia
- Phone - (888)555-1212
- ZIP - 17123
- Country - United States

Other Data

The .us domain is being purchased by a citizen of the U.S., living in the U.S., for personal use.



The values passed for city, state, and country must utilize the case shown above. The API will not process a state value in all capital letters, for example.

XML containing high ASCII characters - accent marks, tildes, etc. - will not be processed by the API.

Using the WAPI web service, this information is defined using the Nexus object.



The product ID values used above can also be found to in the product table in the Product IDs addendum, available for download from the reseller extranet.

The API's return XML contains the ID for the new user it created during the processing of this order request. This value must be captured and persisted for use in the next certification step.

Domain Name Privacy Purchase (Task #3)

Instructions: Use the Poll() method to obtain the resource id of example.biz, then use the OrderDomainPrivacy() method to add privacy to the domain name "example.biz" that was

purchased in the previous step. Create a new privacy account with password "defgh" and email address info@example.biz to manage privacy information.



Privacy accounts are managed by Domains By Proxy® (DBP), a WWD sister company.

Result: The API's return XML should indicate that the privacy request was successfully processed.

Additional Information:

Product ID

- Private registration - 377001

User

A unique identifier was created and returned by the API during the certification task in section 4.1.2. This value must be supplied so the API can create a relationship between the domain shopper and the DBP account created when the privacy is added to the domain.

Domain resource ID

This value is retrieved through the Poll(...) method. Persist the API's return XML from this method, as it will also be used to in the certification step described in 4.1.5.



After the Web Service responds to a poll request, the messages are deleted from the API message queue. It is the reseller's responsibility to persist any data that may be needed for future resource management requests.

Domain Name Availability Check (Task #4)

Instructions: Check the availability of domain names "example.us" and "example.biz".

Result: The return XML should show that neither is available for registration (having been registered earlier in the certification process).

Domain Name Information Query (Task #5)

Instructions: Use the Info() method to query information about the domain name "example.biz", pass the resource id of example.biz.

Result: The return XML should contain the domain name, creation date, owner ID, expiration date, and status.

Domain Name Renewal (Task #6)

Instructions: Use the OrderPrivateDomainRenewals() method to renew the domain names "example.us" and "example.biz" for one additional year. Because privacy has been added to "example.us", this resource will have to be renewed, also.

Result: The return XML should return a message indicating that the two domains and one privacy account were successfully renewed.

Additional Information:

Product IDs

- .US 1-year domain name renewal - 350137
- .BIZ 1-year domain name renewal - 350087
- Privacy 1-year renewal - 387001

User

A unique user identifier was created and returned during the certification task in section 4.1.2. This identifier also appears in the return XML from the certification task described in section 7.1.3

Resource IDs

The resource IDs for both domains were retrieved using the previous call to the Poll() method. The DBP resource ID must be retrieved via a new Poll() method call.

This will return information from the domain name privacy order from the task described in section 4.1.3.

Domain Name Transfer (Task #7)

Instructions: Use the OrderDomainTransfer() method to transfer the domain "example.com" to a new Wild West Domains shopper account.

Result: The return XML should contain a message indicating the order was successfully processed.

Additional Information:

Name Servers

- ns1.example.net, ns2.example.net

Product ID

- .COM Transfer - 350011

Registrant Data

- First Name - Joe
- Last Name - Smith
- Email - joe@smith.us
- Address - 1 S. Main St.
- City - Oakland
- State - California
- ZIP - 97123
- Country - United States
- Phone - (777)555-1212
- Password - "ghijk"

Pass Certification

Upon completion of the above steps, the reseller's status will be updated in the WWD database and a certification confirmation email message will be sent to the reseller. At this point, the reseller is granted permission to the production reseller extranet and API environments.

POST-CERTIFICATION (LIVE ENVIRONMENT)

The transition from OTE to the production API environment, a reseller must make the changes outlined in this section.

Update the Web Reference

In your application project, change the Web Reference URL to point to the production environment (<https://api.wildwestdomains.com/wswwdapi/wapi.asmx?wsdl>).

Also change the account and passwords being used so that it uses the production values.

Update the Reseller Extranet

The IP addresses specified in the OTE extranet do not carry over into the production environment. This information must be updated in the "IP Address Configuration" page (<http://www.resellerextranet.com/account.ipaddresses.aspx>).



The user name for logging in to the production extranet is included in the certification confirmation email and is the user name in the welcome email sent when the API account was purchased.

Wire account funds

Before API purchases can be made, funds must be wired to Wild West Domains. This is accomplished by following the instructions on the following page: <https://www.ote.resellerextranet.com/account/wiretransfer.aspx>.



\$20 fee for Good As Gold deposits greater than \$1000. Resellers can also mail check (subject to 10-day waiting period, but no fees).

WEB SERVICE METHOD FORMAT

Resellers communicate with the WWD API server by invoking the available methods and receiving responses from the API in the XML format.

All methods (except ProcessRequest) takes the form:

```
class WAPI
{
...
String MethodXXX( String clTRID, Credential cred, /*other
parameters*/ )
...
}
```



For a complete list and detailed description of available types and methods offered by the Web service, consult the Wild West Domains Reseller Web Service API documentation (WWDAPI_WebService.pdf).

cITRID (Client Transaction Identifier)

This parameter is used to identify the originating request and subsequent response from the API. It must be a unique value across all pending API requests.



In .NET (or COM), the easiest way to get a unique ID is to generate a GUID (Globally Unique Identifier). Use the `System.Guid.NewGuid()` method to generate a GUID and use it as a unique client transaction ID.

Cred (Credential Object)

This parameter contains the account and password information that the Web service needs to validate (along with the configured IP address) in order to determine whether or not the transaction is legitimate.

RETURN XML FORMAT

The Web service API processes reseller requests and returns XML to be used and/or persisted by the reseller. The format of this XML depends on whether the request was successfully executed.

Response XML Format (Request Success)

A response to a successful request takes the following format:

```
<response cITRID="reseller. 0000000001">
  <result code="1000" />
  <msg><!-- specific api message goes here --></msg>
  <resdata>
    <!-- specific api response goes here -->
  </resdata>
</response>
```

The code attribute value communicates to the reseller whether the request was successful (code="1000"). Data that is pertinent to the specific request appears within the resdata element.

Response XML Format (Request Failure)

A failed request returns XML with the format below:

```
<response>
<result code="1001">
<msg><!-- specific failure message text goes here --></
msg> </result>
</response>
```

A code attribute value other than 1000 (in this case, 1001) signifies a failed request.



For a complete list of code attribute values, see Appendix B:
API code attribute values.

3. Tutorials and Code Samples

The following are instructions on how to reference the Web Service API WSDL and describe how to invoke the available methods.

VISUAL STUDIO .NET

To add a Web reference to the Web service API:

1. Open the project in Visual Studio.
2. In the Solution Explorer window, right-click on the References folder and select Add Web Reference.
3. On the URL combo box, type <https://api.ote.wildwestdomains.com/wswwdapi/wapi.asmx?wsdl> (OTE) or <https://api.wildwestdomains.com/wswwdapi/wapi.asmx?wsdl> (PROD).
4. Change the Web reference name to WsWWDAPI or any name appropriate for the project.
5. Click the Add Reference button.

The project now has access to the types (classes or objects) and methods provided by the Web service.

C# Code Sample: Invoking the Describe Method

```
namespace WAPIClient
{
    class ResellerAPI
    {
        public void Init()
        {
            m_WAPIObj = new WsWWDAPI.WAPI();
            m_Credencial = new WsWWDAPI.Credencial();

            m_Credencial.Account = "account";
            m_Credencial.Password = "password";
        }

        public void CallDescribe()
        {
            String sReturnXml;
            String sCLTRID = System.Guid.NewGuid().ToString();

            sReturnXml = m_WAPIObj.Describe(sCLTRID, m_Credencial);
            Console.WriteLine(sReturnXml);
        }

        [STAThread]
        static void Main(string[] args)
        {
            ResellerAPI reseller = new ResellerAPI();

            reseller.Init();
            reseller.CallDescribe();
        }

        WsWWDAPI.WAPI m_WAPIObj = null;
        WsWWDAPI.Credencial m_Credencial = null;
    }
}
```

VB.NET Code Sample: Invoking the Describe Method

```
Module ResellerAPI

    Sub CallDescribe()

        Dim returnXML As String
        Dim clTRID As String

        clTRID = System.Guid.NewGuid().ToString()
        returnXML = wapi.Describe("TXN.1", credential)
        System.Console.WriteLine(returnXML)

    End Sub

    Sub Main()

        wapi = New WsWWDAPI.WAPI
        credential = New WsWWDAPI.Credential

        credential.Account = "account"
        credential.Password = "password"

        CallDescribe()

    End Sub

    Dim wapi As WsWWDAPI.WAPI
    Dim credential As WsWWDAPI.Credential

End Module
```

PHP Code Sample: Invoking the Describe Method

```
<?PHP
include_once('PEAR.php');
include_once('Net/Socket.php');
include_once('XML/Tree.php');

$dsc_header = array(
    "POST /wswwdapi/wapi.asmx HTTP/1.1",
    "Host: api.ote.wildwestdomains.com",
    "Content-Type: text/xml; charset=utf-8",
    "SOAPAction: \"http://wildwestdomains.com/webserVICES/Describe\"");

$dsc_msg = '<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Describe xmlns="http://wildwestdomains.com/webserVICES/">
      <sCLTRID>reseller.000000001</sCLTRID>
      <credential>
        <Account>api_account</Account>
        <Password>api_password</Password>
      </credential>
    </Describe>
  </soap:Body>
</soap:Envelope>
';

$ch = curl_init("https://api.ote.wildwestdomains.com/wswwdapi/wapi.asmx?WSDL");
if ($ch == FALSE) {
    echo "Connecting to createsend failed\n";
}

curl_setopt($ch, CURLOPT_HTTPHEADER, $dsc_header);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $dsc_msg);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, FALSE);
curl_setopt($ch, CURLOPT_VERBOSE, 0);

$result = curl_exec($ch);
echo "Return XML: \n$result\n";
?>
```

Java Code Sample: Invoking the Describe Method

```
private void describe() throws RemoteException {
    WAPI_Impl wapiImpl = new WAPI_Impl();
    WAPI_Soap wapiSoap = wapiImpl.getWAPI_Soap();
    Credential credential = new Credential();
    credential.setAccount("21320");
    credential.setPassword("hello");
    ((Stub)
wapiSoap)._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY,
"http://172.19.66.39/WSWDDAPI/WAPI.asmx?WSDL");
    String response =
wapiSoap.describe(Guid.NewGuid().ToString().Substring(49),
credential);
    System.out.println(response);
}
```

4. Certification Primers

Certification Primers walk resellers through the tasks that comprise WWD's API certification test. This test ensures a reseller's competency with calling core API methods and his/her ability to query information from the return XML. Primers explain the methods and their parameters; and the XML that is being returned as a result of calling these methods. Utilizing the primers while completing the certification steps will help a reseller quickly master the API's core library of methods.

CERTIFICATION PRIMER - TASK #1 AND #4: CHECKAVAILABILITY

The CheckAvailability() method is used to verify the availability of domain names, name servers, and hosts. For the purposes of certification, it will be used for domain names exclusively.

Table A: CheckAvailability Method Parameters

Parameter	Type	Usage
sCLTRID	String	[Required] Maximum length 50. Must be unique in order. Client transaction identifier that must be unique across all requests.
credential	Credential	[Required] WsWWDAPI.Credential type. This contains the reseller's account id and password used to validate the transaction.
sDomainArray	String[]	[Optional] Checks the availability of the domain names.
sHostArray	String[]	[Optional] Checks the availability of the host names.
sNSArray	String[]	[Optional] Checks the availability of the name servers.

Sample A: CheckAvailability Method Code Snippet

```
public void CheckDomains(String[] sDomainArray)
{
    String sReturnXml;
    String sCLTRID = System.Guid.NewGuid().ToString();

    // Pass in an array of strings i.e. an array of domain
    names
    // to check.
    sReturnXml = m_WAPIObj.CheckAvailability(sCLTRID,
    m_Credential, sDomainArray, null, null);

    //Parse the content of sReturnXml

    ...
}
```

Table B: CheckAvailability Method Return XML elements/attributes

Element/ Attribute	Description
domain	Returns the availability check results. (In this case a domain check.)
name	Specifies the domain name checked.
avail	The availability status of the name. Possible values are: -1 - could not find availability info 0 - domain name not availability for registration 1 - domain name available for registration

Sample B: CheckAvailability Method Return XML

```
<check>
<domain name="x" avail="x" />
</check>
```

CERTIFICATION PRIMER - TASK #2: DOMAIN NAME ORDER REQUEST

The OrderDomains() method is used to submit a domain purchases to the API.

Table A: OrderDomains Method Parameters

Parameter	Type	Usage
sCLTRID	String	[Required] Maximum length 50. Must be unique in order. Client transaction identifier that must be unique across all requests.
credential	Credential	[Required] WsWWDAPI.Credential type. This contains the reseller's account id and password used to validate the transaction.
shopper	Shopper	[Required] WsWWDAPI.Shopper type. The reseller's customer. Pertains to the individual user or customer.
items	DomainRegistration[]	[Required] Array of WsWWDAPI.DomainRegistration objects.
dbpItems	DomainByProxy[]	[Optional] Array of WsWWDAPI.DomainByProxy objects. Note: Match a DomainByProxy object to specific DomainRegistration objects (domains that are registered via a proxy).
sROID	String	[Optional] Maximum length 50 characters. May contain any characters. Optional reseller-supplied order identifier. If given, this value is returned in all notification messages concerning the order.

Table B: Shopper Type Properties

Property	Type	Usage
user	String	[Required] Either createNew or a previously returned user ID. If this value is "createNew," the following non-DBP properties are required. If not, then the value must be a previously returned user ID and all of the following attributes are ignored, even if present.

Property	Type	Usage
pwd	String	<p>[Optional] Minimum 5 characters, maximum 20 characters.</p> <p>Required if user="createNew." This is the password that is to be assigned to the new account.</p> <p>If "user" refers to an existing account ID, this field is ignored even if present.</p>
pwdhint	String	<p>[Optional] Maximum 256 characters. A password hint that is displayed to the user upon request on the Web site. This field is always optional.</p> <p>If user="createNew," this field is assigned as the password hint to the new user account.</p> <p>If this field is not present, the account will not have a password hint. If user refers to an existing account ID, this field is ignored if present.</p>
email	String	<p>[Optional] Maximum 80 characters.</p> <p>The user's email address. Must be in a valid email address format. For example, a@b.c.d.com or a@b.us.</p> <p>If user="createNew," this field is required.</p> <p>If user refers to an existing user ID, this field is ignored.</p>
firstname	String	<p>[Optional] Maximum 30 characters. The user's first name.</p> <p>If user="createNew," this field is required.</p> <p>If user refers to an existing user ID, this field is ignored.</p>
lastname	String	<p>[Optional] Maximum 50 characters. The user's last name.</p> <p>If user="createNew", then this field is required.</p> <p>If user refers to an existing user ID, then this field is ignored.</p>

Property	Type	Usage
phone	String	<p>[Optional] Format: +[0-9]{1,3}\.[0-9]{1,12}</p> <p>Example: +1.4805058800</p> <p>The user's phone number. If user="createNew," this field is required.</p> <p>If user refers to an existing user ID, this field is ignored.</p>
dbpuser	String	<p>[Optional] createNew or a previously returned dbpuser ID. User ID for the dbp account. If a domainByProxy item (privacy renewal) is present or a resourceRenewal with productid 387001, this field is required, otherwise it is ignored.</p> <p>If dbpuser="createNew", then a new dbp account is created using the following 3 attributes.</p> <p>If its value is not "createNew", then a check is made to ensure that the given dbpuser ID is valid.</p>
dbppwd	String	<p>[Optional] Minimum 5 characters, maximum 20 characters.</p> <p>If a domainByProxy item is present in the order, this field is required, otherwise it is ignored.</p> <p>If dbpuser="createNew," this value is assigned as the password to the newly created account.</p> <p>If dbpuser refers to an existing dbpuser ID, then this value must be the password of that account.</p> <p>If the password doesn't match, the order will be rejected.</p>
dbppwdhint	String	<p>[Optional] Maximum 256 characters.</p> <p>Password hint for the new dbp account. This field is always optional.</p> <p>If a domainByProxy item is present and dbpuser="createNew," this value is assigned to the new dbp account.</p> <p>If not present, the account will not have a password hint.</p> <p>If dbpuser refers to an existing account ID, this field is ignored.</p>

Property	Type	Usage
dbpemail	String	[Optional] Maximum 80 characters. Email account used to send dbp-related email to the user. Required only if a domainByProxy item is present and dbpuser="createNew".

Table C: DomainRegistration Type Properties

Property	Type	Usage
order	OrderItem	[Required] WsWWDAPI. OrderItem type. This contains the order information. Refer to the OrderItem type.
sld	String	[Required] Maximum 63 characters. Second level domain name (abc of abc.com).
tld	String	[Required] (.com, .net, .org, .us, .ws, or .info) Top level domain (com of abc.com).
period	Int32	[Required] Length of the registration, in years. Valid values for most are 1-10.
registrant	ContactInfo	[Required] The registrant contact. Refer to the ContactInfo type.
nexus	Nexus	[Optional] Refer to the Nexus type.
nsArray	NS[]	[Optional] Array of NS objects.
admin	ContactInfo	[Optional] The admin contact. Refer to the ContactInfo type.
billing	ContactInfo	[Optional] The billing contact. Refer to the ContactInfo type.
tech	ContactInfo	[Optional] The tech contact. Refer to the ContactInfo type.
autorenew-flag	Int32	[Optional] Default value is 1. Supply 1 to auto-renew; 0 for manual renew.

Table D: OrderItem Type Properties

Property	Type	Usage
productid	Int32	[Required] The WWD product ID from the catalog of the item being purchased.
quantity	Unsigned Int32	[Optional] The quantity of the item being purchased (defaults to 1). Must be a positive integer.
riid	String	[Optional] Maximum length 50. May contain any characters. Optional reseller-supplied item identifier. If given, this value will be returned in all notification messages sent to the reseller.
duration	Double	[Optional] Floating point value; default=1.0 The duration of the purchase. This attribute is used only on domainByProxy items. If privacy is being purchased at the same time that the domain name is being registered, the duration attribute in the domainByProxy item must match the period attribute in the domainRegistration node. If privacy is being purchased for an already-registered domain name, then use the info request to retrieve the proper value for this attribute.

Table E: ContactInfo Type Properties

Property	Type	Usage
fname	String	[Required] Maximum 30 characters. Contact's first name. Not required if "org" has a value.
lname	String	[Required] Maximum 50 characters. Contact's last name. Not required if "org" has a value.
org	String	[Optional] Name of the organization. This may be given in place of fname, lname.
email	String	[Required] Maximum 80 characters. Must be in a valid email address format. For example, a@b.c.d.com or a@b.us.
sa1	String	[Required] Street address. Maximum 30 characters. Pattern: ^[a-zA-Z0-9 #&'()+,.-./:;@[\]]+\$

Property	Type	Usage
sa2	String	[Optional] Street address 2. Maximum 30 characters. Pattern: <code>^[a-zA-Z0-9 #&'()+,.-./:;@\[\]]+\$</code>
city	String	[Required] Maximum 30 characters. City of residence.
sp	String	[Required] Maximum 30 characters. State or province. Required if cc="United States" or cc="Canada," must be valid state or province name (full name).
pc	String	[Required] Postal code. Maximum length 10 characters. Pattern: <code>^[a-zA-Z0-9 #&'()+,.-./:;@\[\]]+\$</code>
cc	String	[Required] Country of residence. Must match one of the entries in the Countries table (spelled out name).
phone	String	[Required] Phone number. Format: <code>+ [0-9]{1,3} \. [0-9]{1,12}</code> Example: +1.4805058800
fax	String	[Optional] Fax number. Format: <code>+ [0-9]{1,3} \. [0-9]{1,12}</code> Example: +1.4805058800

Sample A: OrderDomains Method Code Snippet

```
public void OrderDomains()
{
    String sReturnXml;
    String sCLTRID = System.Guid.NewGuid().ToString();

    WsWWDAPI.Shopper shopper = new WsWWDAPI.Shopper();
    shopper.user = "123456";
    shopper.pwd = "password";

    WsWWDAPI.ContactInfo registrant = new
    WsWWDAPI.ContactInfo();
    registrant.fname="John";
    registrant.lname="Doe";
    registrant.org="Wild West Reseller";
    registrant.email="j.doe@coolexample.com";
    registrant.sa1="123 Sample St.";
    registrant.sa2="Suite 1";
    registrant.city="Scottsdale";
    registrant.sp="Arizona";
    registrant.pc="85260";
    registrant.cc="United States";
    registrant.phone="+1.4805058857";
    registrant.fax="+1.4808241499";

    //Register just one domain.
    WsWWDAPI.DomainRegistration[] domainRegArray = new
    WsWWDAPI.DomainRegistration[1];
    domainRegArray[0] = new WsWWDAPI.DomainRegistration();
    domainRegArray[0].order = new WsWWDAPI.OrderItem();
    domainRegArray[0].order.productid = 350030;
    domainRegArray[0].order.quantity = 1;
    domainRegArray[0].order.riid = "1";
    domainRegArray[0].order.duration = 1;
    domainRegArray[0].sld = "mysampledomain";
    domainRegArray[0].tld = "net";
    domainRegArray[0].period = 1;
    domainRegArray[0].registrant = registrant;
    domainRegArray[0].admin = registrant;
    domainRegArray[0].billing = registrant;
    domainRegArray[0].tech = registrant;

    sReturnXml = m_WAPIObj.OrderDomains(sCLTRID, m_Credential,
    shopper, domainRegArray, null, "1");

    //Parse the content of sReturnXml
    ...
}
```


Sample B: OrderDomains Method Return XML

```
<result code="1000">
  <msg>processed n i t e m s</msg>
</result>
<resdata>
  <orderid>x</orderid>
</resdata>
```

Table F: OrderDomains Method Return XML Elements/Attributes

Element/Attribute	Description
msg	This string should reflect that the domain name registration has been processed.
orderid	The unique order identifier assigned to by the API.

CERTIFICATION PRIMER - TASK #3: PRIVACY ORDER REQUEST

The OrderDomainPrivacy() method is used to add privacy to an existing domain name registration.

NOTE: Privacy may also be added at the same time the domain name is registered using the OrderDomains() method by passing in the dbplItems parameter.

Table A: OrderDomainPrivacy Method Parameters

Parameter	Type	Usage
sCLTRID	String	[Required] Maximum length 50. Must be unique in order. Client transaction identifier that must be unique across all requests.
credential	Credential	[Required] WsWWDAPI.Credential type. This contains the reseller's account ID and password used to validate the transaction.
shopper	Shopper	[Required] WsWWDAPI.Shopper type. The reseller's customer. Pertains to the individual user or customer.
dbplItems	Domain-ByProxy[]	[Required] Array of WsWWDAPI.DomainByProxy objects. Note: Match a DomainByProxy object to specific DomainRegistration objects (domains that are registered via a proxy).
sROID	String	[Optional] Maximum length 50 characters. May contain any characters. Optional reseller-supplied order identifier. If given, this value is returned in all notification messages concerning the order.

Table B: DomainByProxy Type Properties

Parameter	Type	Usage
sld	String	[Required] Maximum 63 characters. Second level domain name (abc of abc.com).
tld	String	[Required] (.com, .net, .org, .us, .ws, or .info). Top level domain (com of abc.com).
resourceid	String	[Optional] The resource ID returned in a previous notification message associated with the original order for the domain name being renewed.

Sample A: OrderDomainPrivacy Method Code Snippet

```
public void AddPrivacyToDomain(String sId, String tId)
{
    String sReturnXml;
    String sCLTRID = System.Guid.NewGuid().ToString();

    WsWWDAPI.Shopper shopper = new WsWWDAPI.Shopper();
    shopper.user = "123456";
    shopper.pwd = "password";

    WsWWDAPI.DomainByProxy[] dbpArray = new
    WsWWDAPI.DomainByProxy[1];
    dbpArray[0] = new WsWWDAPI.DomainByProxy();
    dbpArray[0].order = new WsWWDAPI.OrderItem();
    dbpArray[0].order.productid = 377001;
    dbpArray[0].order.quantity = 1;
    dbpArray[0].order.riid = "1";
    dbpArray[0].order.duration = 1;
    dbpArray[0].sId = sId;
    dbpArray[0].tId = tId;

    sReturnXml = m_WAPIObj.OrderDomainPrivacy(sCLTRID,
    m_Credential, shopper, dbpArray, "1");

    //Parse the content of sReturnXml

    ...
}
```

Sample B: OrderDomainPrivacy Method Return XML

```
<response user="x" dbpuser="x" svTRID="x" clTRID="x">
<result code="1000">
<msg>processed x item(s)</msg>
</result>
<resdata>
<orderid>x</orderid>
</resdata>
</response>
```

Table C: OrderDomainPrivacy Method Return XML Elements/Attributes

Element/ Attribute	Description
user	This is a pass-through value supplied by the reseller and pass-back by the API.
dbpuser	The unique identifier assigned to the account by WWD.
orderid	The unique identifier assigned by the API to the transaction.

CERTIFICATION PRIMER - TASK #5: DOMAIN INFO REQUEST

The Info() method is used to get information about items that have been previously ordered. Up to 100 items may be sent in a single info request.

Table A: Info Method Parameters

Parameter	Type	Usage
sCLTRID	String	[Required] Maximum length 50. Must be unique in order. Client transaction identifier that must be unique across all requests. Suggested format: GUID or UUID; Globally Unique Identifier or Universally Unique Identifier (e.g. "F9168C5E-CEB2-4FAA-B6BF-329BF39FA1E4").
credential	Credential	[Required] WsWWDAPI.Credential type. This contains the reseller's account id and password used to validate the transaction.
sResourceID	String	[Optional] The resource id of the item for which details are requested. This parameter is optional only if either domain or order id is provided. Specifies the resource whose information will be retrieved, but only if the resource is owned by the logged in reseller.
sType	String	<p>[Optional] The type of information being requested. Default value = standard.</p> <p>standard - Returns the standard information regarding the resource. What is standard varies from resource to resource.</p> <p>name - Applicable to domain resources only. This value returns the domain name.</p> <p>createDate - Returns the creation date of the resource (e.g., when it was purchased).</p> <p>ownerID - Returns the shopper id of the owner of the resource.</p> <p>expirationDate - Returns the expiration date of the resource.</p> <p>status - Returns the status of the resource (this is not the order status, but indicates if the particular resource is active or not). This status values vary from resource to resource.</p>

Parameter	Type	Usage
		<p>maxRenewYears - Applicable to domain resources only. This value returns the number of years for which a particular domain may be renewed up to a maximum of 10.</p> <p>dbpDuration - Applicable to domain resources only. Returns the proper duration value for a Domains By Proxy® order that is adding privacy to an existing domain name.</p> <p>access - Applicable to domain resources only. This value returns a list of the shopper IDs that have management rights to this resource.</p> <p>autoRenewDate - Returns the auto renew date of the resource. If an empty string is returned, the resource will not be auto-renewed.</p> <p>dns - Applicable to domain resources only. This value returns the DNS resource record information from the zone file associated with the given domain.</p> <p>contactEmails - Applicable to domain resources only. This value returns the contact email addresses for the following contacts: registrant, admin, and tech.</p>
sDomain	String	[Optional] The domain of the item for which details are requested. This parameter is optional only if either resource id or order id is provided. Information about the domain, as specified by the sType parameter, is returned, but only if the specified domain was registered by the logged in reseller.
sOrderID	String	[Optional] The order id of the item for which details are requested. This parameter is optional only if either resource id or domain is provided. Information about the order status of all items in the order will be returned, but only if the order id given was placed by the logged in reseller.

Sample A: Info Method Code Snippet

```
public void GetDomainInfo(String domain)
{
    String sReturnXml;
    String sCLTRID = System.Guid.NewGuid().ToString();

    sReturnXml = m_WAPIObj.Info(sCLTRID, m_Credential, null,
    null, domain, null);

    //Parse the content of sReturnXml

    ...
}
```

Sample B: OrderDomainPrivacy Method Return XML

```
<result code="1000"/>
<resdata>
<info resourceid="domain: XXXX" name="x. TLD"
createDate="x" ownerId="x" expirationDate="x"
status="0" />
</resdata>
</response>
```

NOTE: The API return XML for any resource info request will contain information relevant to the resource as well as pass through the resourceid originating from the info request.

CERTIFICATION PRIMER - TASK #6: DOMAIN RENEWAL ORDER REQUEST

The OrderDomainRenewals() method is used for renewing registration for an existing domain name. In cases where the domain being renewed has privacy assigned to it, the reseller needs to call the OrderPrivateDomainRenewals method.

Table A: OrderDomainRenewals Method Parameters

Parameter	Type	Usage
sCLTRID	String	[Required] Maximum length 50. Must be unique in order. Client transaction identifier that must be unique across all requests.
credential	Credential	[Required] WsWWDAPI.Credential type. This contains the reseller's account ID and password used to validate the transaction.
shopper	Shopper	[Required] WsWWDAPI.Shopper type. The reseller's customer. Pertains to the individual user or customer.
items	DomainRenewal[]	[Required] Array of WsWWDAPI.DomainRenewal objects.
sROID	String	[Optional] Maximum length 50 characters. May contain any characters. Optional reseller-supplied order identifier. If given, this value is returned in all notification messages concerning the order.

Table B: DomainRenewal Type Properties

Parameter	Type	Usage
resourceid	String	[Required] The resource ID returned in a previous notification message associated with the original order for the domain name being renewed.
sld	String	[Required] Maximum 63 characters. Second level domain name (abc of abc.com).
tld	String	[Required] (.com, .net, .org, .us, .ws, or .info). Top-level domain (com of abc.com).
period	Int32	[Required] Length of the registration, in years. Valid values for most are 1-10.

Sample A: OrderDomainRenewals Method Code Snippet

```
public void RenewDomain(String sId, String tId)
{
    String sReturnXml;
    String sCLTRID = System.Guid.NewGuid().ToString();

    WsWWDAPI.Shopper shopper = new WsWWDAPI.Shopper();
    shopper.user = "123456";
    shopper.pwd = "password";

    WsWWDAPI.DomainRenewal[] domainRenewArray =
    new WsWWDAPI.DomainRenewal[1];

    domainRenewArray[0] = new WsWWDAPI.DomainRenewal();
    domainRenewArray[0].order = new WsWWDAPI.OrderItem();
    domainRenewArray[0].order.productId = 350041;
    domainRenewArray[0].order.quantity = 1;
    domainRenewArray[0].order.riid = "1";
    domainRenewArray[0].order.duration = 1;
    domainRenewArray[0].resourceid = "domain: 1519008";
    domainRenewArray[0].sId = sId;
    domainRenewArray[0].tId = tId;
    domainRenewArray[0].period = 1;

    sReturnXml = m_WAPIObj.OrderDomainRenewals(sCLTRID,
    m_Credential, shopper, domainRenewArray, "1");

    //Parse the content of sReturnXml

    ...
}
```

Sample B: OrderDomainRenewals Method Return XML

```
<response user="x" svTRID="x" clTRID="x">
<result code="1000">
<msg>processed x item(s)</msg>
</result>
<resdata>
<orderid>x</orderid>
</resdata>
</response>
```

CERTIFICATION PRIMER - TASK #7: DOMAIN TRANSFER ORDER REQUEST

The OrderDomainTransfers() method is used for transferring ownership of a domain from one user to another.

Table A: OrderDomainTransfers Method Parameters

Parameter	Type	Usage
sCLTRID	String	[Required] Maximum length 50. Must be unique in order. Client transaction identifier that must be unique across all requests.
credential	Credential	[Required] WsWWDAPI.Credential type. This contains the reseller's account ID and password used to validate the transaction.
shopper	Shopper	[Required] WsWWDAPI.Shopper type. The reseller's customer. Pertains to the individual user or customer.
items	Domain-Transfer[]	[Required] Array of WsWWDAPI.DomainTransfer objects.
sROID	String	[Optional] Maximum length 50 characters. May contain any characters. Optional reseller-supplied order identifier. If given, this value is returned in all notification messages concerning the order.

Table B: DomainTransfer Type Properties

Property	Type	Usage
order	OrderItem	[Required] WsWWDAPI.OrderItem type. This contains the order information. Refer to the OrderItem type.
sld	String	[Required] Maximum 63 characters. Second-level domain name (abc of abc.com).
tld	String	[Required] (.com, .net, .org, .us, .ws, or .info). Top level domain (com of abc.com).
authInfo	String	[Optional] Used by some registries as a means of validating transfer requests. Specifically, .us, .biz and .info

Sample A: OrderDomainTransfers Method Code SnippetL

```
public void TransferDomain(String sId, String tId)
{
    String sReturnXml;
    String sCLTRID = System.Guid.NewGuid().ToString();

    WsWWDAPI.Shopper shopper = new WsWWDAPI.Shopper();
    shopper.user = "123456";
    shopper.pwd = "password";

    WsWWDAPI.DomainRenewal[] domainRenewArray =
    new WsWWDAPI.DomainRenewal[1];

    WsWWDAPI.DomainTransfer[] domainXferArray =
    new WsWWDAPI.DomainTransfer[1];

    domainXferArray[0] = new WsWWDAPI.DomainTransfer();
    domainXferArray[0].order = new WsWWDAPI.OrderItem();
    domainXferArray[0].order.productId = 350160;
    domainXferArray[0].order.quantity = 1;
    domainXferArray[0].order.riid = "1";
    domainXferArray[0].order.duration = 1;
    domainXferArray[0].sId = sId;
    domainXferArray[0].tId = tId;
    domainXferArray[0].authInfo = "123451";

    sReturnXml = m_WAPIObj.OrderDomainTransfers(sCLTRID,
    m_Credential, shopper, domainXferArray, "1");

    //Parse the content of sReturnXml

    ...
}
```

Sample B: OrderDomainTransfers Method Return XML

```
<result code="1000">
<msg>processed x item(s)</msg>
</result>
<resdata>
<orderid>x</orderid>
</resdata>
```

A. Poll Requests

The Poll() method provides the reseller with the messages regarding pending orders. Receiving a success after submitting an Order method indicates the order was successfully submitted but is in a pending state and has to be fulfilled in our system. You should call the Poll method once a day to receive the status of your orders and the resource IDs for each of the domains/resources in an order.

Table A: Poll Method Parameters

Parameter	Type	Usage
sCLTRID	String	[Required] Maximum length 50. Must be unique in order. Client transaction identifier that must be unique across all requests.
credential	Credential	[Required] WsWWDAPI.Credential type. This contains the reseller's account ID and password used to validate the transaction.
sOp	String	[Optional] The only valid value for this attribute is "req." which is the default value. This requests that pending notifications be delivered.

Sample A: Poll Method Code Snippet

```
public void TransferDomain(String sId, String tId)
{
    String sReturnXml;
    String sCLTRID = System.Guid.NewGuid().ToString();

    sReturnXml = m_WAPIObj.Poll(sCLTRID, m_Credential, "req");


    //Parse the content of sReturnXml

    ...
}
```

Sample B: Poll Method Return XML

```
<result code="1004">
<msg>messages waiting</msg>
</result>
<msgQ count="4" date="x" />
<resdata>
<REPORT>
<ITEM orderid="x" roid="" riid="wwd.0" status="1"
timestamp="x" />
<ITEM orderid="x" roid="" riid="wwd.1" status="1"
timestamp="x" />
<ITEM orderid="x" roid="" riid="wwd.0"
resourceid="domain: xxxx" status="2" timestamp="x" />
<ITEM orderid="x" roid="" riid="wwd.1"
resourceid="domain: xxxx" status="2" timestamp="x" />
</REPORT>
</resdata>
</response>
```

Table B: Poll Method Return XML elements/attributes

Element/ Attribute	Description
orderid	WWD's unique order identifier.
roid	A return of the optional value supplied in a previous OrderXXX method call made by the reseller.
Riid	The row (line-item) identifier.
status	Status can take the following values: 1 - delivered 2 - processed 3 - cancelled 4 - renewed 5 - auto-renew failed 6 - auto-renew off 7 - auto-renew on 19 - transfer away 20 - resource id change 96 - invalid details 97 - insufficient funds 98 - not processed 99 - invalid product 999 - error
	Retrieving messages via a poll request result in the deletion of the retrieved messages. It is the responsibility of the reseller to persist this information.

B. API Return Code Values

Every request sent by the reseller triggers a corresponding response from the API.

Table A: Response Codes - General

Code Value	Description
1000	Success
1001	Failure
1002	Request too big

Table B: Response Codes - Credential Authentications

Code Value	Description
1500	Login Denied; account at connection limit
1501	Login Denied; invalid account IP address
1502	Login Denied; account inactive

Table C: Response Codes - Poll

Code Value	Description
1003	No messages waiting
1004	Messages waiting

Table D: Response Codes - Database Error Code Values

Code Value	Description
2000	Empty
2001	Required field(s) missing
2002	Pattern matching error
2003	Field too long
2004	DB matching error

C. Web Service Methods Overview

The WWD Web Service API exposes the following Web methods below (Table A).

Table A: Web Service Methods

Method	Description
Cancel	Used for canceling existing resources.
CheckAvailability	Checks the availability of domains, hosts and name servers.
CheckUser	Used for checking whether the account information for the specified user is valid or not.
CreateNewShopper	Used for creating a new reseller sub-account.
Describe	Returns activity timeout and version information from the server.
DomainForwarding	Used for forwarding a domain to a specified URL.
GetDomainAlertCredits	Used for querying the number of domain alert, private backorder, and backorder credits.
GetExpiringNameList	This method is used to get the list of expiring domain names for users that have an Investors Edge subscription.
GetMonitoredDomainList	Used for querying a list of monitored domains.
Info	Gets information about items that have been previously ordered.
ManageTransfer	Used for managing the transfer of domains.
NameGen	Used to get the list of alternative domain names based on a given name.
NameGenDB	Used to get the list of alternative domain names based on a given name (Domains Bot Service).

Method	Description
NameGenDBWithTimeLimit	Used to get the list of alternative domain names based on a given name (Domains Bot Service).
OrderCredits	Used to place an order on items that do not require customization (non-domain e.g. email accounts, etc.).
OrderDomainBackOrders	Used for backordering an existing (public or private) domain name.
OrderDomainPrivacy	Used for adding privacy to a domain registration.
OrderDomainRenewals	Used for renewing registration for an existing domain name.
OrderDomains	Used for registering new domain names.
OrderPrivateDomainRenewals	Used for renewing registration for existing private and/or public domain names.
OrderDomainTransfers	Used for transferring ownership of a domain from one user to another.
OrderResourceRenewals	Used for renewing non-domain items.
Poll	Used to retrieve status notifications about pending orders.
ProcessRequest	Provide support for calling WWD services using request XMLs.
RemoveDomainAlert	Used for removing an existing domain alert record.
ResetPassword	Used for resetting the user's password in the WWD system.
SetDomainLocking	Used for setting or clearing the lock on a list of domains.
SetShopperInfo	Used for allowing the reseller to modify the user information for either their top-level account or any of its sub-accounts.
SetupDomainAlert	Used for setting domain alert credit or backorder a domain.
UpdateDomainAlert	Used for updating an existing domain alert record.
UpdateDomainContact	Used for modifying associated contact information for the given domains.

Method	Description
UpdateDomainForwarding	Used for performing multiple domain forwarding.
UpdateDomainMasking	This method is used for performing multiple domain masking.
UpdateDomainOwnership	Used for changing the ownership of a resource from one end user to another (both must be end-users of the currently logged in reseller).
UpdateNameServer	Used for modifying associated name servers for the given domains.
ValidateRegistration	This method is used for validating a domain's period and contacts.