Query as a Web Service Guide

BusinessObjects XI Release 2

Windows and UNIX

Contents

Overview of Query as a Web Service

Query as a Web Service lets you create custom web services for specific queries using Business Objects Web Services. You access queries to build applications that use the same semantic layer as the rest of the BusinessObjects suite.

Business Intelligence (BI) content is usually bound to a specific user interface of BI tools. Query as a Web Service changes this by allowing BI content to be delivered to any user interface that can process Web Services.

Using Query as a Web Service, business users define their own query from a universe, and then easily and securely publish that query as a standalone web service.

Query as a Web Service provides new types of user-driven client solutions for businesses. For example, it enables Crystal Xcelsius to aggregate multiple disparate data sources into a trusted BI view.

Query as a Web Service also enables a range of client-side solutions in tools such as:

- Microsoft Office, Excel, and InfoPath
- SAP NetWeaver
- OpenOffice
- Business rules and process management applications
- Enterprise Service Bus platforms

Note: Business Objects provides a wide range of Web Services for developers. Developers use these Web Services in IDEs with languages such as C# and Java. For more information, go to:

http://www.businessobjects.com/products/dev_zone/

Architecture

Query as a Web Service is based on the W3C web service specifications SOAP, WSDL, and XML. It has two main components:

- Server component

  The server component (included in BusinessObjects XI R2 SP2) stores the Query as a Web Service catalog and hosts the published Web Services.

- Client tool

  This is where business users create and publish Query as a Web Service. You can install the client tool on several machines that can then access and share the same Query as a Web Service catalog stored on the server. The client tool communicates with the server components via Web Services.

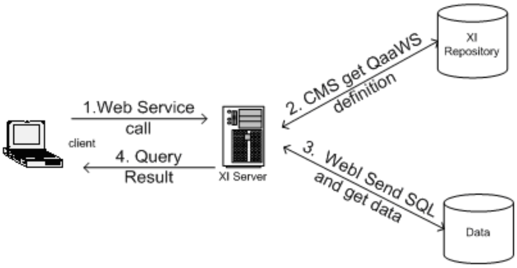


Figure 2-1 Query as a Web Service architecture diagram

Installation prerequisites

Before installing the software, read the Terms of Use agreement and conditions.

The following must be installed on the server:

- BusinessObjects XI Release 2 SP2 Enterprise Edition
- Web Services
- Tomcat or another supported Web Application and JDK

  For an updated list of supported Web Applications and versions, see:

  http://support.businessobjects.com/supported_platforms_xi_release2/

The following must be installed on the client before you install Query as a Web Service:

- .NET 1.1

The client platform must be supported for use with BusinessObjects XI Release 2. For an updated list of supported versions, see:

http://support.businessobjects.com/supported_platforms_xi_release2/

Installing Query as a Web Service

The server component of Query as a Web Service is installed automatically when you install BusinessObjects Enterprise XI Release 2 Service Pack 2 with Web Intelligence.

You must then install the client tool on every machine that will access the server via Web Services.

You must have a separate license for the client tool. It is not covered by the general license for BusinessObjects XI Release 2 Service Pack 2.

▶ **To install the client tool**

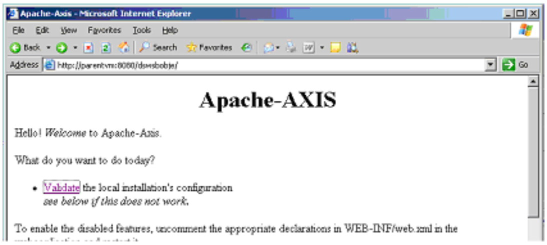
1. Navigate to the Add-Ons\Query as a Web Service folder on the BusinessObjects Enterprise collaterals CD or locate the Query as a Web Service setup.exe file on your network.
2. Double-click setup.exe to launch the Query as a Web Service Wizard.
3. Follow the on-screen instructions in the Query as a Web Service Installation Wizard to complete the installation procedure.

Validating the server component installation

There are few simple steps to follow to verify your installation is correct.

▶ **To validate the Query as a Web Service server component installation**

1. Open a browser and type the following URL `http://[server]:[port]/dswsbobje`, where `[server]` corresponds to your web server address and `[port]` corresponds to the port where you set up in your web server. The default used by Tomcat is 8080. `dswsbobje` is the default name of the Web Service provider web application.


Apache-Axis - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go

http://parentvm:8080/dswsbobje/

Apache-AXIS

Hello! *Welcome* to Apache-Axis.

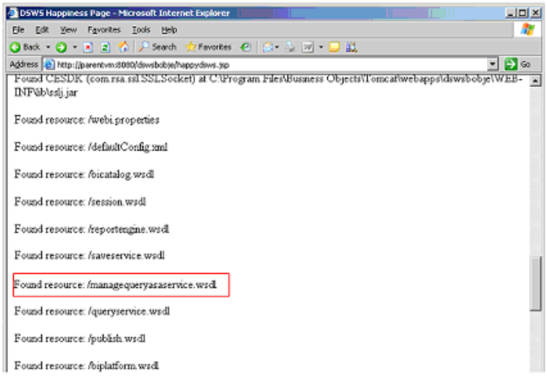
What do you want to do today?

- Validate the local installation's configuration
  see below if this does not work.

To enable the disabled features, uncomment the appropriate declarations in WEB-INF/web.xml in the

2. Click on Validate link to get a status of the installation.

3. Browse down to make sure Query as a Web Service is installed. You will see Found resource:/managequeryasaservice.wsdl


DSWS Happiness Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://parentvm:8080/dswsbobje/happydsws.jsp
Go

```
Found CESDK (com.rsa.ssl.SSLSocket) at C:\Program Files\Business Objects\Tomcat\webapps\dswsbobje\WEB-INF\lib\sslj.jar

Found resource: /webi.properties

Found resource: /defaultConfig.xml

Found resource: /bicatalog.wsdl

Found resource: /session.wsdl

Found resource: /reportengine.wsdl

Found resource: /saveservice.wsdl

Found resource: /managequeryasaservice.wsdl

Found resource: /queryservice.wsdl

Found resource: /publish.wsdl

Found resource: /biplatform.wsdl
```

4. In your browser, now type the following URL: `http://[server]:[port]/dswsbobje/qaawsservices`, where `[server]` corresponds to your server address and `[port]` corresponds to the port where you set up your web server.

   Your browser page appears and contains the title “Available QaaWS services”. This page contains a list of the Query as a Web Service that have already been created on this system.

Troubleshooting

Here are a few tips to follow in the event that you have installation issues.

Launching the CMS first

During launch of Tomcat, in the initialization of the Servlet, QaaWS cache reads the definition of the QaaWS in the repository to build its cache. Thus, ensure that the CMS is launched before Tomcat. If you launch the CMS automatically via the NT Services, you will not need to do this manually.

Setting CMS name

By default, the Web Services connects to the local machine name's CMS. If you want to change to a dedicated CMS, you must change the domain property in the `dsws.properties` file.

1. Stop Tomcat.
2. Open the installation path folder of XIR2: `[InstallationPath]\Tomcat\webapps\qaaws\WEB-INF\classes`

3. Open the file `dsws.properties` and locate:

   `domain=`

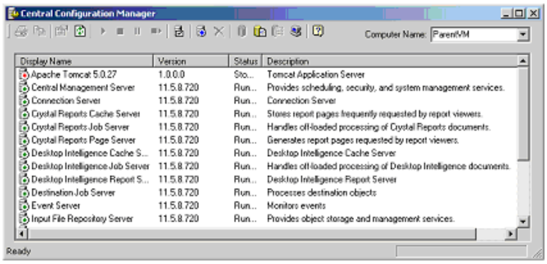
4. Enter your CMS name.
5. Close and save the `dsws.properties` file.
6. Start Tomcat.
7. Check everything is OK by following the section, "Validating the server component installation" on page 7.

Activate the web service provider traces in a test environment

If there is configuration issue, the Business Objects Administrator may be required to set up traces to allow better troubleshooting.

Note: Business Objects strongly recommends using traces only for troubleshooting purposes in test environments.

1. In the Central Configuration Manager (CCM), stop "Apache Tomcat 5.0.27" service.



Computer Name: ParentVM

Display Name	Version	Status	Description
Apache Tomcat 5.0.27	1.0.0.0	Sto...	Tomcat Application Server
Central Management Server	11.5.8.720	Run...	Provides scheduling, security, and system management services.
Connection Server	11.5.8.720	Run...	Connection Server
Crystal Reports Cache Server	11.5.8.720	Run...	Stores report pages frequently requested by report viewers.
Crystal Reports Job Server	11.5.8.720	Run...	Handles off-loaded processing of Crystal Reports documents.
Crystal Reports Page Server	11.5.8.720	Run...	Generates report pages requested by report viewers.
Desktop Intelligence Cache S...	11.5.8.720	Run...	Desktop Intelligence Cache Server
Desktop Intelligence Job Server	11.5.8.720	Run...	Handles off-loaded processing of Desktop Intelligence documents.
Desktop Intelligence Report S...	11.5.8.720	Run...	Desktop Intelligence Report Server
Destination Job Server	11.5.8.720	Run...	Processes destination objects
Event Server	11.5.8.720	Run...	Monitors events
Input File Repository Server	11.5.8.720	Run...	Provides object storage and management services.

Ready

2. Change the trace level.

   By default Query as a Web Service only traces errors. You may be requested to provide additional traces for customer assurance.

3. Edit `log4j.properties` located in `[installationpath]\dswsbobje\WEB-INF\classes\`
4. Type the following in the properties file:

```
    log4j.logger.com.businessobjects=DEBUG, BO1
```

5. Change the trace location.

   By default, it traces in the output console output. If you want to trace a file, comment the `ConsoleAppender` and uncomment the `RollingFileAppender`. If Tomcat is set as a service, you will trace in `dswsbobje.log` under `C:\WINDOWS\system32.`:

```
# console appender
# log4j.appender.BO1=org.apache.log4j.ConsoleAppender
#
    log4j.appender.AXIS1=org.apache.log4j.ConsoleAppen
    der

# rolling file appender
log4j.appender.BO1=org.apache.log4j.RollingFileAppend
    er
log4j.appender.BO1.File=dswsbobje.log
log4j.appender.BO1.Append=false
log4j.appender.BO1.MaxBackupIndex=5
log4j.appender.BO1.MaxFileSize=10
```

6. In the CCM, start "Apache Tomcat 5.0.27" service.

Removing the client tool

To remove the client tool, go to the Windows Control Panel, click **Add or Remove Programs**, then click, **Query as a Web Service**.

Configuring Query as a Web Service Client to connect to a reverse proxy web server

Reverse proxy is a network address translation of a machine from a URL in a given network to a URL in another external network. As Query as a Web Service client binds to Report Engine, Query and BICatalog Web Service you must specify the external URL of the Web Services

For example if your external is URL was http://uws.businessobjects.com/dswsbobje/

You must update the following properties in the dsws.properties file

This file is located in dswsbobje web application.

```
wsresource1=ReportEngine|reportengine web service
alone|http://[myserver.mycompany.com]/dswsbobje/services/
reportengine

wsresource2=BICatalog|bicatalog web service alone|http://
[myserver.mycompany.com]/dswsbobje/services/bicatatog

wsresource4=QueryService|query web service alone|http://
[myserver.mycompany.com]/dswsbobje/services/query
```

Setting Query as a Web Service background user

Query as a Web Service definitions are stored in the repository. Query as a Web Services Web Application needs a BusinessObjects account to handle WSDL generation. You define this account using the following parameters

```
qaaws.principal.username=QaaWSOnly
qaaws.principal.password= QaaWSOnlyPassword
qaaws.principal.authentication=secLAP
```

To avoid security risk, create a dedicated user that does not have access to documents or any other BI content.

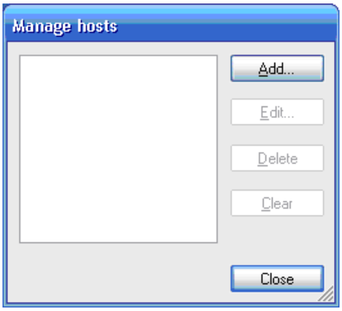
Launching the client tool

▶ **To launch the Query as a Web Service client tool**

- In the Windows Start menu, point to Programs > Business Objects XI Release 2 > BusinessObjects Enterprise > Query As A Web Service.

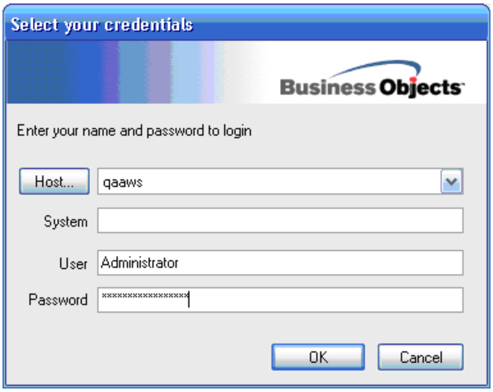
  Either of the following occur:

  - If you are logging in for the first time, the Manage hosts dialog box appears.



Go to "Logging in for the first time" on page 12.

- If this is not your first login, the Select your credentials dialog box appears.



Enter your name and password to login

System

User Administrator

Password

Host: qaaws

Enter your password and click **OK**. Go to “Creating and Publishing a Query As a Web Service” on page 14.

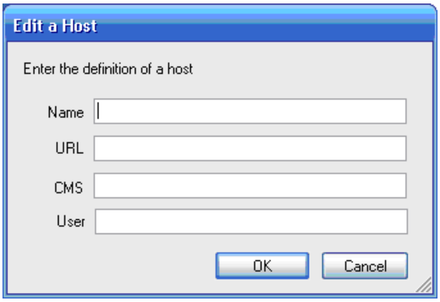
Logging in for the first time

The first time you log in, you need to define the system host where the web services are installed. At subsequent logins, you only need to enter your password.

▶ **To log into the client tool for the first time**

1. In the Manage hosts dialog box, click **Add**.

   The Edit a Host dialog box appears.



Enter the definition of a host

Name

URL

CMS

User

2. Type a system name in the **Name** field.

   The URL is automatically entered.

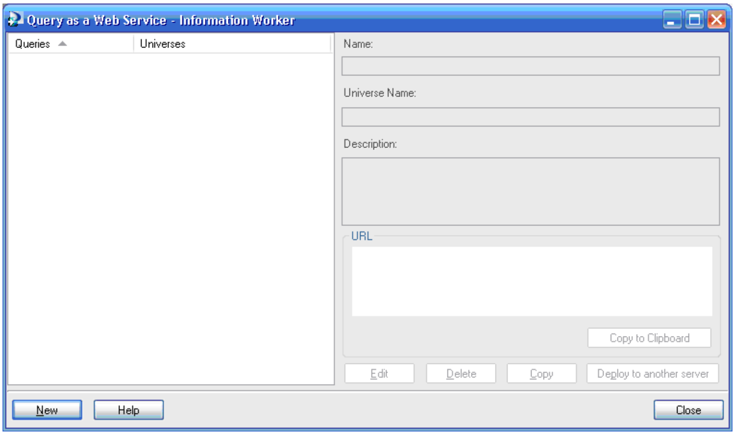
3. Make sure the port is correct.

4. If the CMS is not on the same server as the Web Services, complete the CMS field.
5. Enter a user name that will be used by default, then click **OK**.

   The Manage hosts dialog box reappears and the host name appears in the list.
6. Click **OK**.

   The Select your credentials dialog box appears.
7. Enter your password, then click **OK**.

   The Query as a Web Service client tool appears.

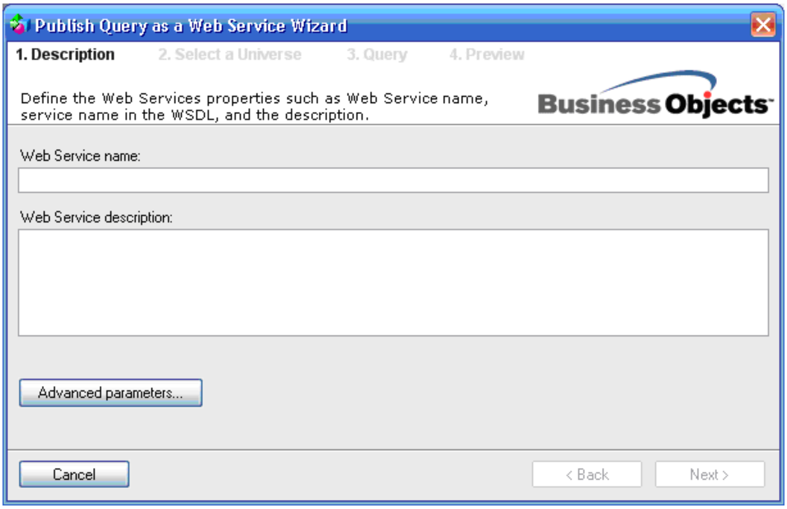


Creating and Publishing a Query As a Web Service

▶ **To create and publish a Query as a Web Service**

1. Open the Query as a Web Service client tool.

   If you have not yet published a Query as a Web Service, the Publish Query as a Web Service Wizard appears.



Define the Web Services properties such as Web Service name, service name in the WSDL, and the description.

Web Service name:

Web Service description:

   - If you have already published a Query as a Web Service, click **New** to access the wizard.

2. In the **Web Service name** field, enter a name for the Query as a Web Service.

   The characters are restricted to avoid problems during WSDL generation.

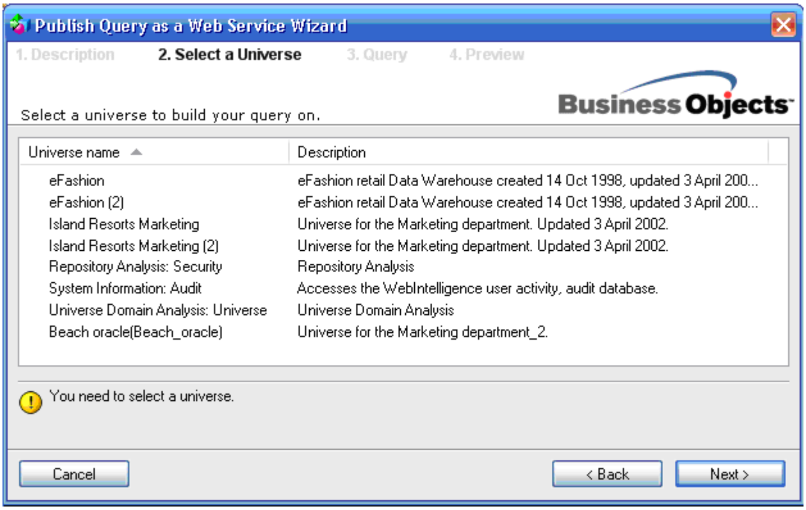
3. In the Web Service Description box, enter a description that will help you and others re-use the query.

4. If you want to set advanced parameters, click **Advanced Parameters.**

   - For instructions, see “Setting advanced parameters” on page 25.

5. Click **Next.**

   The Select a Universe page appears.



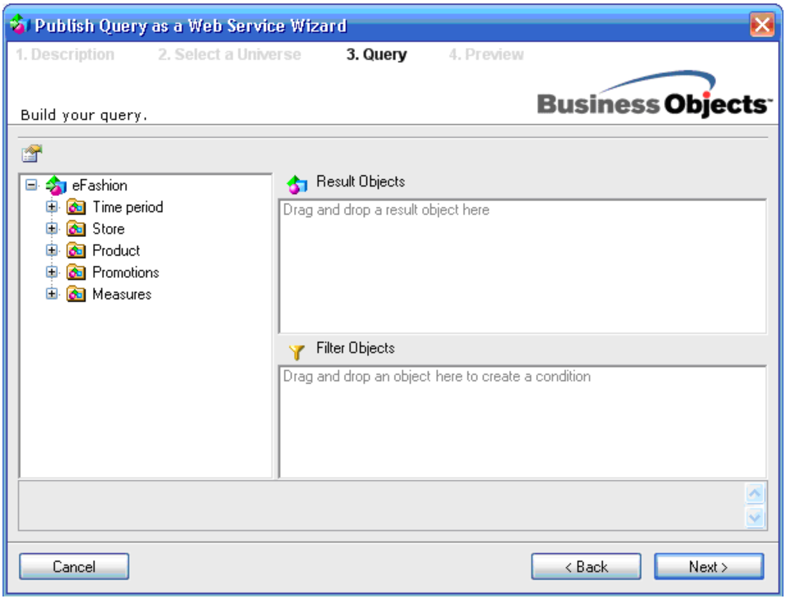
Select a universe to build your query on.

Universe name	Description
eFashion	eFashion retail Data Warehouse created 14 Oct 1998, updated 3 April 200...
eFashion (2)	eFashion retail Data Warehouse created 14 Oct 1998, updated 3 April 200...
Island Resorts Marketing	Universe for the Marketing department. Updated 3 April 2002.
Island Resorts Marketing (2)	Universe for the Marketing department. Updated 3 April 2002.
Repository Analysis: Security	Repository Analysis
System Information: Audit	Accesses the WebIntelligence user activity, audit database.
Universe Domain Analysis: Universe	Universe Domain Analysis
Beach oracle(Beach_oracle)	Universe for the Marketing department_2.

You need to select a universe.

6. Select the universe from which you want to create the query, and then click **Next**.

   The Query Panel appears.



Build your query.

Drag and drop a result object here

Drag and drop an object here to create a condition

7. Create your query.

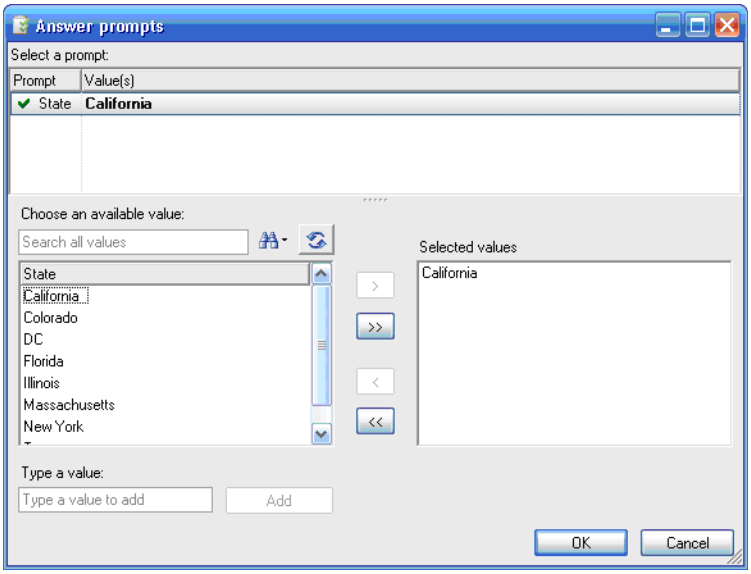
   For information on query creation, see the *Building Queries Using Web Intelligence Query - HTML* guide.

8. Click **Next.**

Creating a prompt

You can create a prompt in a query to refine the query results.

1. Select a prompt object, then click the right arrows until the value appears in the Selected values pane.



Prompt	Value(s)
✓ State	**California**

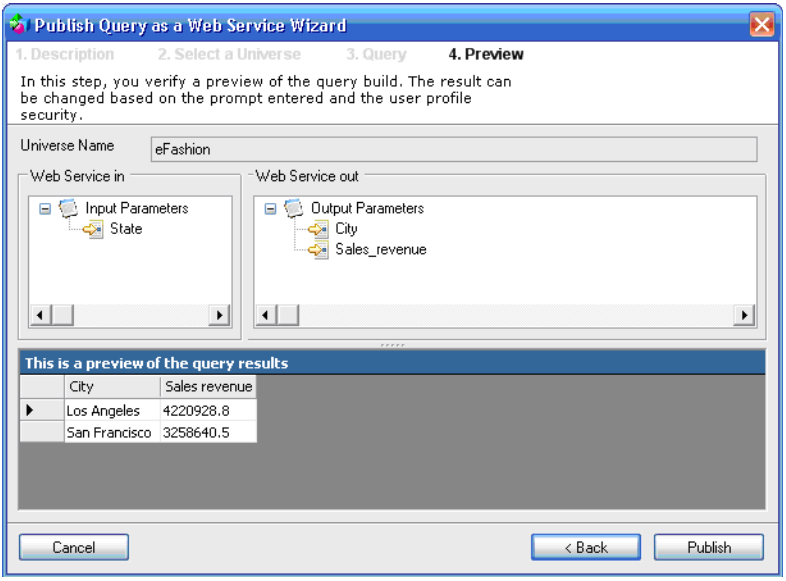
Choose an available value:

State
California
Colorado
DC
Florida
Illinois
Massachusetts
New York

Selected values
California

Type a value:
Type a value to add

2. Review the query prompts and contexts, if necessary, before displaying the Preview page. For more information on contexts see, “Using contexts with Query as a Web Service” on page 18.



Publish Query as a Web Service Wizard

1. Description   2. Select a Universe   3. Query   **4. Preview**

In this step, you verify a preview of the query build. The result can be changed based on the prompt entered and the user profile security.

Universe Name: eFashion

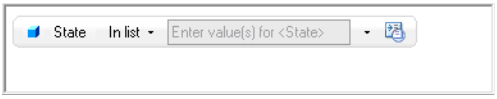
Web Service in
- Input Parameters
  - State

Web Service out
- Output Parameters
  - City
  - Sales_revenue

This is a preview of the query results

City	Sales revenue
Los Angeles	4220928.8
San Francisco	3258640.5

- The **Web Service in** pane contains the input parameters of the Web Services you create. Each input parameter is linked to a filter prompt that you created in the Filter Objects pane of the Query Panel.



State   In list   Enter value(s) for <State>

Tip: You can change the default value of the prompt by clicking the prompt icon.

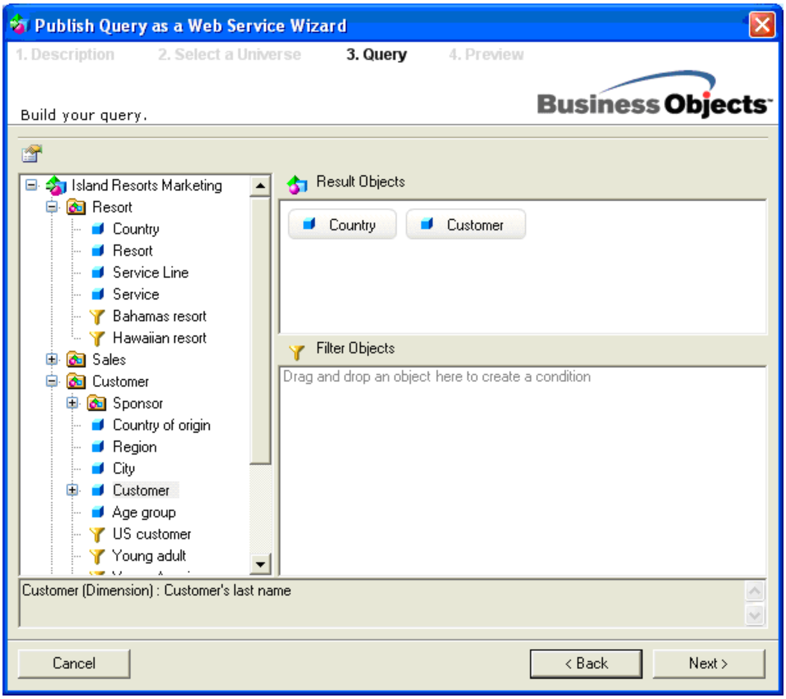
Note: The name of the Prompt text is important because it will appear in the list of input parameters of the query web service at runtime. The prompt text appears in the Wizard, Step 4. Preview.

- The **Web Service out** pane contains the output parameters of the Web Services you create. The output parameters are linked to Result Objects you put in the Query Panel; they are expressed as a result set.

Using contexts with Query as a Web Service

You can define a Query that uses the same context. You select the context at design-time and always reuse that context at run-time. Follow the example below to learn how to create a query with a context.

1. Define a Query that requires the selection of a context, such as Sales or Reservations in the Island Resorts marketing database.

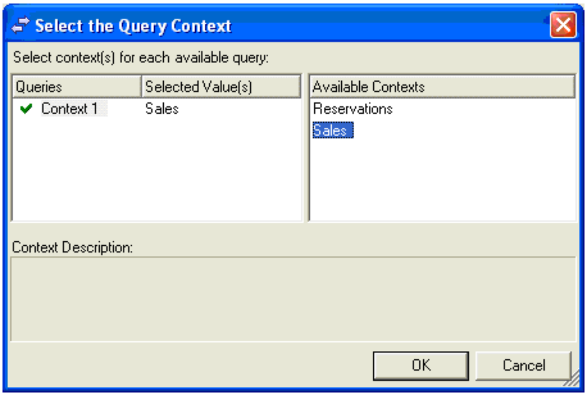


Build your query.

Drag and drop an object here to create a condition

Customer (Dimension) : Customer's last name

2. Click **Next** to display the Select the Query Context window.



Select the Query Context

Select context(s) for each available query:

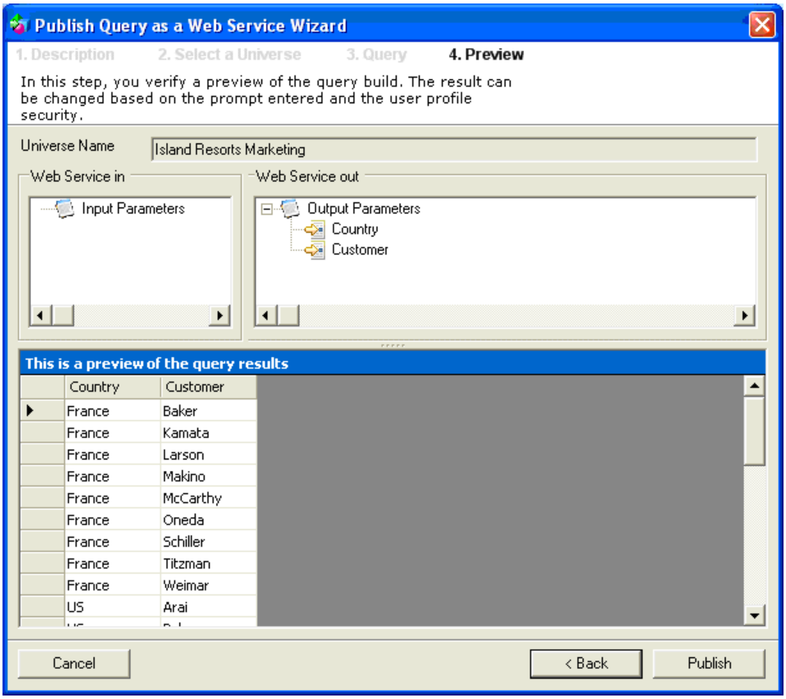
Queries	Selected Value(s)
✓ Context 1	Sales

Available Contexts
Reservations
Sales

Context Description:

3. Of the two available contexts "Reservations" and "Sales", select "Sales".

4. Click OK, to display the Preview pane.



Publish Query as a Web Service Wizard

1. Description   2. Select a Universe   3. Query   **4. Preview**

In this step, you verify a preview of the query build. The result can be changed based on the prompt entered and the user profile security.

Universe Name: Island Resorts Marketing

Web Service in
- Input Parameters

Web Service out
- Output Parameters
  - Country
  - Customer

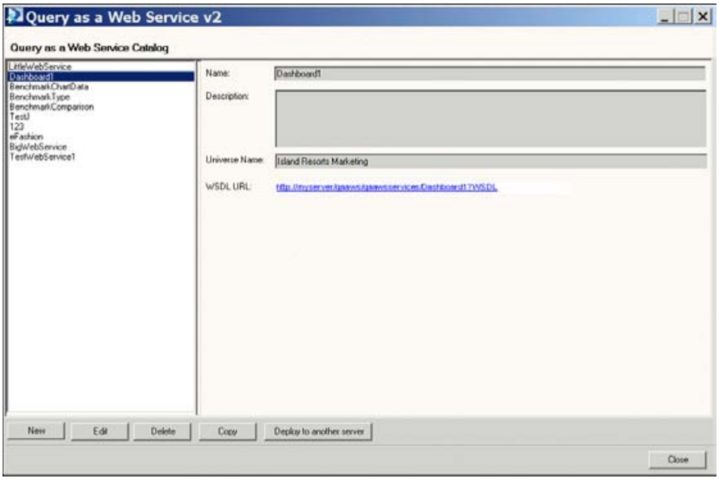
This is a preview of the query results

Country	Customer
France	Baker
France	Kamata
France	Larson
France	Makino
France	McCarthy
France	Oneda
France	Schiller
France	Titzman
France	Weimar
US	Arai

Note: The two contexts do not appear in the Input Parameters list. Once you select the context in the query design step, the selected context is stored in the query definition. At runtime, you cannot select a different context than the one that is saved at design time — in our example "Sales".

Managing Query as a Web Service

When you launch the Query as a Web Service client tool, the Query Catalog appears on the left side. It shows the queries you created.



LittleWebService
Dashboard1
BenchmarkChartData
BenchmarkType
BenchmarkComparison
TestJ
123
eFashion
BigWebService
TestWebService1

Name: Dashboard1

Description:

Universe Name: Island Resorts Marketing

WSDL URL: http://myserver/qaaws/qaawsservices/Dashboard1?WSDL

In addition to viewing each query property, you can perform the following actions on your queries:

- Edit
- Delete
- Copy
- Deploy to another server

These actions are explained in the sections below.

Viewing Query as a Web Service properties

▶ **To view the properties of a Query as a Web Service**

1. In the Query Catalog, select the Query as a Web Service you want.

   The properties display on the right side of the panel.

2. To open the WSDL in a web browser, click the WSDL URL.

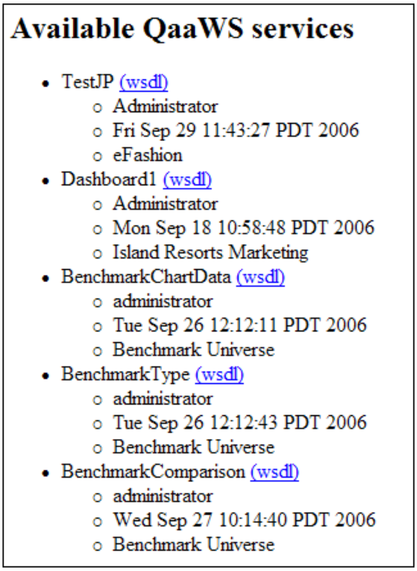
Viewing queries in a web browser

You can also view, in a web browser, your instances of Query as a Web Service.

▶ To view available web services in a web browser

- Open a web browser, and go to the following URL:dswsbobje

```
http://<name of server>:<Tomcat port number>/dswsbobje/
    qaawsservices
```



Available QaaWS services

- TestJP (wsdl)
  - Administrator
  - Fri Sep 29 11:43:27 PDT 2006
  - eFashion
- Dashboard1 (wsdl)
  - Administrator
  - Mon Sep 18 10:58:48 PDT 2006
  - Island Resorts Marketing
- BenchmarkChartData (wsdl)
  - administrator
  - Tue Sep 26 12:12:11 PDT 2006
  - Benchmark Universe
- BenchmarkType (wsdl)
  - administrator
  - Tue Sep 26 12:12:43 PDT 2006
  - Benchmark Universe
- BenchmarkComparison (wsdl)
  - administrator
  - Wed Sep 27 10:14:40 PDT 2006
  - Benchmark Universe

Editing a Query as a Web Service

Editing a Query as a Web Service may change the associated WSDL, and this could break the link that is used by others to communicate with Web Services. It is therefore recommended that you notify users of any changes you make.

▶ To edit a Query as a Web Service

1. In the Query Catalog, select the Query as a Web Service you want to edit.

2. Click **Edit**.

   The Query as a Web Service appears in the Publish Query as a Web Service Wizard.

3. Make the changes you want.

   For instructions, see "Creating and Publishing a Query As a Web Service" on page 14.

Deleting a Query as a Web Service

▶ **To delete a Query as a Web Service**

1. In the Query Catalog, select the Query as a Web Service you want to delete.
2. Click **Delete**.

Copying a Query as a Web Service

You must change the name of the Query as a Web Service when you copy it.

▶ **To copy a Query as a Web Service**

1. In the Query Catalog, select the Query as a Web Service you want to copy.
2. Click **Copy**.

   The Query as a Web Service appears in the Publish Query as a Web Service Wizard.

   The default name of the copied query is `Copy_<name of original>`.

3. Change the default name, if you want.
4. Pass through the remaining pages of the wizard and, if you want, modify the information.

Deploying queries to another server

This section explains how to deploy to another server; that is, copy a Query as a Web Service definition from one server to another. For example, you can move a definition from a development server to a test or production server.

Before starting, make sure the universe and users are the same on both machines. Use the Import Wizard or BIAR files to import universes and users. It is important to have the same CUID during Import Wizard operations.

To deploy to another server, you can use:

- the Query as a Web Service client tool

  An advantage of this method is that the query will be automatically pointed to the web server on the new system.
- the Import Wizard
- a BIAR file

Deploying using the Query as a Web Service client tool

▶ **To deploy to another server using the client tool**

1. In the Query Catalog, select the Query as a Web Service whose definition you want to copy.
2. Click **Deploy**.

   The Select Your Credentials dialog box appears.
3. Complete the information for the system on which you want to deploy the Web Services, and then click **OK**.

   The Query as a Web Service appears in the Publish Query as a Web Service Wizard.
4. Publish the Query as a Web Service to the new system.

   For instructions, see "Creating and Publishing a Query As a Web Service" on page 14.

Note: Deploying a Query as a Web Service definition to another server automatically changes the WSDL location and the services execution location. You can customize this service-based URL using the wizard. You can also make it dynamic in an Xcelsius project by using the Input Values text box of the Web Services Connectivity. Using this functionality, you can switch a dashboard from development to production by changing this URL.

Deploying using Import Wizard

To deploy using the Import Wizard, import the Query as a Web Service definition from the source server to the destination server.

▶ **To deploy to another server using the Import Wizard**

1. Make sure the Import Wizard is installed on both the source and destination server.
2. In the CCM of the source server, stop the Central Management System service.
3. In the Business Objects installation directory of the server, find and open the QueryAsAWebService.zip file.

4. In the ImportWizard folder, find and copy the qaaws.dll file.
5. Paste the file into:

   ```
   $INSTALLDIR\BusinessObjects Enterprise
      11.5\win32_x86\plugins
   ```
6. Obtain a RegSvr.reg registry entry from either:
   - http://www.kinook.com/Download/RegSvr.reg
   - http://www.devx.com/tips/Tip/19957
7. In the Windows Explorer, right-click the qaaws.dll file and point to **Register COM Server**.

   A confirmation message appears.
8. Click **OK**.
9. Restart the CMS.
10. Repeat the above procedure on the destination server.
11. On the destination server, use the Import Wizard to import the Query as a Web Service defintion from the source server.

    Query as a Web Service definitions are considered by the wizard to belong to the object type “application folders and objects.”
12. After importing, point the newly-deployed query definition to the web server on the destination system.

Deploying using a BIAR file

Before you begin:

- Make sure the Import Wizard is installed on both the source and destination server.
- Follow the procedure in “Deploying using Import Wizard” on page 23 to copy and register the qaaws.dll file.

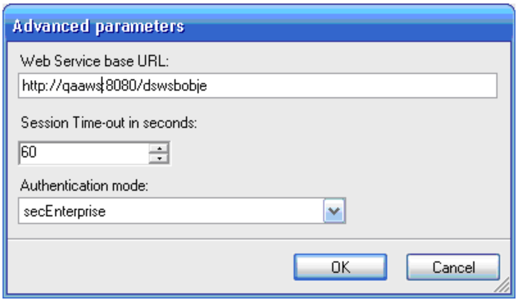
▶ **To deploy to another server using a BIAR file**

1. Open the Import Wizard on the source server.
2. After logging in to the source CMS, select the target BIAR file to which you will export the definitions.
3. In the Select Objects to Import dialog box, select **Import folders and objects** > **Import application folders and objects**.
4. In the Select Application Folders and Objects dialog box, select the Query as a Web Service definitions you want.
5. Continue through the remaining steps of the Import Wizard.

6. Open the Import Wizard on the destination server.
7. In the Source Environment dialog box, select the BIAR file to which you exported the definitions.
8. Log into the destination CMS.
9. In the Select Objects to Import dialog box, select **Import folders and objects** > **Import application folders and objects**.
10. In the Select Application Folders and Objects dialog box, select the Query as a Web Service definitions you want.
11. Continue through the remaining steps of the Import Wizard.
12. After importing, point the newly-deployed query definition to the web server on the destination system.

Setting advanced parameters

You can set several advanced parameters. To access the dialog box click the **Advanced parameters** button on the bottom left of the 1. Description page of the Publish Query as a Web Service Wizard.



Web Service base URL:
http://qaaws:8080/dswsbobje

Session Time-out in seconds:
60

Authentication mode:
secEnterprise

Web Service base URL

Reverse proxy is a network address translation of a machine from a URL in a given network to a URL in another network, usually an external network like the public internet.

Example: Using the base URL technique

A server called `myserver.company.com` within a company network could be called: `www.mycompany.com` in the external network.

To support such a deployment, you must set up a Web Services base URL. The base URL contains the external URL from which you want your Web Service to be accessible, for example `www.mycompany.com/dswsbobje/`.

To use Query as a Web Service client tool with a reverse proxy Web Server, see "Configuring Query as a Web Service Client to connect to a reverse proxy web server" on page 10.

Session time-out in seconds

To improve the performance of Query as a Web Service, particularly the cascading call scenario, the user's connection to the server is cached by the web service provider. You can configure session time-out (in seconds) for each Query as a Web Service connection. The default is 60 seconds.

For example, if a given user login calls *service 1* and then under 60 seconds calls *service 2* with the same login (identical username and password), the server reuses the same connection and reinitializes the session time-out.

Authentication mode

Authentication mode is the type of directory against which the BusinessObjects XI R2 platform validates the login. Examples include Enterprise, LDAP, Windows AD, and SAP.

You can set the authentication mode so that it will be defined according to the service, or by the consumer:

- Service

  You, as administrator, select the authentication directory; all users subsequently accessing the service authenticate on this directory (except for the sessionID option).

  All authentication directories supported by the server are available for selection in the Authentication Mode drop-down list.

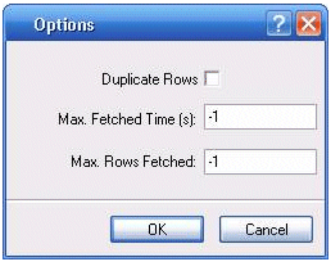
- Consumer defined

  The consumer of the query selects the authentication mode as an input parameter called authenticationType.

Query Panel options

You can set several options for the Query Panel in the Options dialog box.

To access the dialog box, click the **Options** icon in the Query Panel of the Publish Query as a Web Service Wizard.



Duplicate Rows □

Max. Fetched Time (s): -1

Max. Rows Fetched: -1

Duplicate rows

By default, the query will not return duplicate rows (this is different than in earlier versions). If you want duplicate rows, select this check box.

Reset context on refresh

Clear this check box if you don’t want the context to be displayed.

Max. fetched time

Enter the maximum date fetch time, in seconds. The default value “-1” means that this option is deactivated and the universe connection will be taken into account.

Max. row fetched

Enter the maximum number of rows to fetch. The default value “-1” means that this option is deactivated and the universe connection will be taken into account.

Consuming a Query as a Web Service

There are several ways to consume a Query as a Web Service:

- via WSDL
- in Crystal Xcelsius
- in Crystal Reports
- in Microsoft Office InfoPath

WSDL

WSDL is an XML-based description of how to communicate using the web service; that is, the protocol bindings and message formats required to interact with the Web Services listed in its directory.

The supported operations and messages are described on a high level and then bound to a concrete network protocol and message format. WSDL is often used in combination with SOAP and XML schema to provide Web Services via the Internet.

A client program connecting to a web service can read the WSDL to determine what functions are available on the server.

To find the WSDL for a Query as a Web Service, select it in the Query Catalog.

Crystal Xcelsius

To consume a Query as a Web Service inside Crystal Xcelsius, use the Web Service Connector.

The Web Service Connector component allows a Flash document created in Xcelsius to communicate with Query as a Web Service via SOAP, using point and click. The Xcelsius Flash document is self-contained and communicates with the web service to display data visually. The only prerequisite is that there be a SOAP-based web service available to the Flash document.

The Web Service Connector component, when activated, creates a SOAP-based message (basically an XML document) and sends it to the web service. The web service responds with a SOAP-based message of its own. The Web Service Connector component then sends this data to all the other components, resulting in a live visual representation of your data.

There are many public Web Services available, and many different toolkits and packages for SOAP-based Web Services. Using public Web Services and packages that already have a web service on top is easy. You need only a WSDL document for the web service. For Query as a Web Service, you can find the WSDL in the properties of each web service by selecting it in the Query Catalog.

See the *Crystal Xcelsius User Guides* for more information.

Cross-domain issue

After downloading the Xcelsius widget via the web, you may encounter difficulty retrieving data with the Query as a Web Service client tool if the Flash and the client tool come from different web domains.

This occurs for security reasons related to Macromedia Flash. The Flash displayed in a browser is not permitted to access data residing outside the web domain from which the Flash file format (SWF) originated.

The solution depends on whether your Xcelsius server and the Query as a Web Service client tool are on the same or different machines.

Same machine

Open the Optional Parameters dialog box (see “Web Service base URL” on page 25), and modify the Web Service Base URL so that it matches the web domain from which you downloaded Xcelsius.

Different machines

For instructions, go to:

http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_14213

Selecting the web service

Selecting the web service involves pointing the Web Service Connector component to the WSDL document.

▶ **To select the web service**

1. When working on an Excel spreadsheet, double-click the Web Service Connector component to open the Properties panel.
2. Click **Select Web Service**.

   The Select Web Service dialog box appears.
3. In the WSDL URL box, type or paste the location of the WSDL document.
4. Click **Submit**.

   If there was an error, the WSDL document may not be fully validated or it may not meet requirements of the Web Service Connector component.

5. In the Methods list, select the method you want to bind with.

   For a given web service, you can bind to only one method per component.

6. Click **OK**.

   The input and output messages are now available for you to tie to your data.

Input messages

For input messages, only elements can be tied to data.

Use the “**-**” button to remove folders and elements. This prevents the folder or field from being sent in the message.

The “**+**” button can be used to add a folder or a repeating element.

Output messages

For output messages, both elements and folders can be tied to data.

Selecting a folder displays the number of columns in that folder. When you tie this to data, each element in the folder is assigned to a column in the order the elements appear.

Selecting fewer columns limits the data that is bound to the number of columns that you select. Selecting additional columns inserts blank columns. If an element is repeating underneath the folder, only the first element will be mapped to the column. The folders underneath the selected folder cannot be mapped.

Use the “**-**” button to remove unnecessary elements. This contracts the view of the tree and may reduce processing time in the Flash document.

Authentication in Xcelsius

Xcelsius provides an authentication mechanism that enables you to avoid logging into InfoView twice with the same session ID.

Keep in mind the following rules if you customize authentication:

- An existing session ID is used only if the user name and password are blank (if they are not hard coded or not passed as input values captured by a dialog). This occurs when Query as a Web Service is running in InfoView or Dashboard Manager.
- If the user name and password are *not* blank, then use these values to authenticate the user. No session is created. This is the most scalable scenario and is the preferred option for large-scale usage.

- If there is no pre-existing session (and username and password are blank), Xcelsius displays the standard security dialog, in which a session is created. This occurs when Xcelsius designers do not build their own security dialog.

Best practices

Best practises when using Xcelsius and Query as a Web Service are different than those frequently used for reporting.

In reporting, queries often fetch hundreds or thousands of rows. In addition, Business Objects reporting products display content as well as store and process it.

This is not the case with Xcelsius, which is a visualization tool that requires all data it displays to be delivered to it, presentation-ready, by the queries it runs.

As a result, here are some best practices for building Xcelsius solutions that fully leverage BusinessObjects Enterprise and the semantic layer:

- Fetch only the data you need for the display component, at that particular point in the users viewing sequence, and no more. Do not “overfetch.”
- Fetch the data you need at the level of aggregation you need it at, and not lower.
- Highly parameterize and/or highly aggregate every query in the model in order to reduce result set size.
- Enable Drilling to Detail by passing the parent selection into a child query that uses the selection as input.
- Build more queries containing fewer rows and columns, instead of the opposite (building fewer queries containing more rows and columns). Models perform better as a result.
- Perform the bulk of the calculation logic in the database or universe, not the Excel layer. Models perform better as a result.
- Optimize the underlying database for pure query speed, modeling it for the dashboard solution it needs to support. If the database is too slow, build another, better and faster one (or use a cube).
- Do not retrieve more than 500 rows in a single Query as a Web Service. Flash cannot handle result sets much larger than this. Find a way to parametrize the query.
    - If requirements call for more than 500 rows, form an OpenDoc URL and use a URL Link component to pass the request to a prompted Web Intelligence or Crystal Report.

Crystal Reports

This section explains how Crystal Reports can consume Query as a Web Service as a data source.

▶ **To consume a Query as a Web Service in Crystal Reports**

1. In the Crystal Reports Standard Report Creation Wizard, on the Data page, create a new XML connection.
2. In the XML Type and Location page of the XML dialog box, select *Use Web Server Data Source*, and then click **Next**.

   The Web Services Location page appears.
3. In the HTTP WSDL URL field, type the WSDL of the selected Query as a Web Service.

   The Authentication page appears.
4. Set Basic authentication (if you haven't already), and then click **Next**.

   The Web Service, Port, and Method page appears.
5. Complete the information, and then click **Finish**.

   The Enter Values dialog box appears.
6. Set the Web Services parameters with login, password, and prompts, and then click **OK**.

   The Data page of the Standard Report Creation Wizard re-appears.
7. Select *runQueryAsServiceResponse/table/row*. and then click **Next**.

   The Fields page appears.
8. Select the field to build your query on, and then click **Next**.

   A report is created.
9. Refresh the report.

   The correct parameters are shown in the report.

Microsoft Office InfoPath

This section explains how Microsoft Office InfoPath can consume Query as a Web Service as a data source.

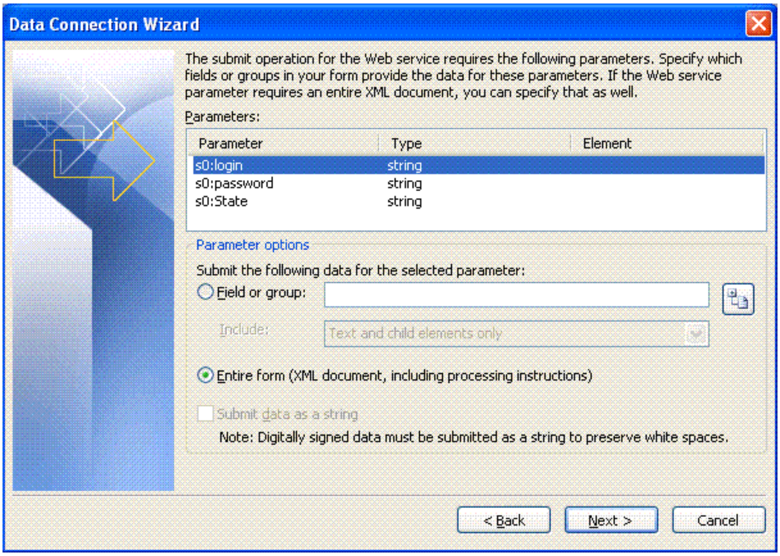
▶ **To consume a Query as a Web Service in InfoPath**

1. In InfoPath, access the Design a Form task list.
2. Click **New From Data Connection**.

   The Data Connection Wizard appears.

3. Select **Web Service**, and then click **Next**.
4. Select **Receive and Submit Data**, and then click **Next**.
5. Type or browse for the WSDL file, and then click **Next**.
6. Select the web service operation, and then click **Next.**
7. Enter a name for the data connection, and then click **Next**.
8. Type or browse for the web service you want users to submit their forms to, and then click **Next**.

   The Parameters page appears.



Data Connection Wizard

The submit operation for the Web service requires the following parameters. Specify which fields or groups in your form provide the data for these parameters. If the Web service parameter requires an entire XML document, you can specify that as well.

Parameters:

Parameter	Type	Element
s0:login	string	
s0:password	string	
s0:State	string	

Parameter options

Submit the following data for the selected parameter:

○ Field or group:

Include: Text and child elements only

● Entire form (XML document, including processing instructions)

□ Submit data as a string

Note: Digitally signed data must be submitted as a string to preserve white spaces.

9. For each parameter, select **Entire Form**, and then click **Next**.
10. Type a name for the data connection submitting data, and then click **Next**.

    The data form appears on the left, and the data source on the right.
11. Build the form, and then click **Run Query**.

Limitations

You may encounter certain limitations when using Query as a Web Service. These can occur when creating a query, or at run-time.

When creating a query

- Multi-cubes cannot be used
- Combined queries and subqueries cannot be used
- IndexAware prompts are not implemented

Also, keep in mind that the name of the Web Service and its metadata is encoded to support various programming languages (such as C#, Java, C++, VB, Flash).

At run-time

- Object restrictions cannot be used