IBM

U2 Web Services Developer

Version 10.3
February, 2009

IBM Corporation
555 Bailey Avenue
San Jose, CA 95141

AIX, DB2, DB2 Universal Database, Distributed Relational Database Architecture, NUMA-Q, OS/2, OS/390, and OS/400, IBM Informix®, C-ISAM®, Foundation.2000 ™, IBM Informix® 4GL, IBM Informix® DataBlade® module, Client SDK™, Cloudscape™, Cloudsync™, IBM Informix® Connect, IBM Informix® Driver for JDBC, Dynamic Connect™, IBM Informix® Dynamic Scalable Architecture™ (DSA), IBM Informix® Dynamic Server™, IBM Informix® Enterprise Gateway Manager (Enterprise Gateway Manager), IBM Informix® Extended Parallel Server™, i.Financial Services™, J/Foundation™, MaxConnect™, Object Translator™, Red Brick® Decision Server™, IBM Informix® SE, IBM Informix® SQL, InformiXML™, RedBack®, SystemBuilder™, U2™, UniData®, UniVerse®, wIntegrate® are trademarks or registered trademarks of International Business Machines Corporation.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Windows, Windows NT, and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names used in this publication may be trademarks or service marks of others.

This product includes cryptographic software written by Eric Young (eay@cryptosoft.com).

This product includes software written by Tim Hudson (tjh@cryptosoft.com).

Documentation Team: Claire Gustafson, Shelley Thompson, Anne Waite

Table of Contents

Preface

IBM U2 Web Services Developer is designed for customers to provide a simple and easy environment to publish their UniVerse database resource as web services, with minimum knowledge of XML, SOAP, WSDL, and so forth.

The IBM U2 Web Services Developer Tool allows you to select different database resources, and publish them as web services, such as subroutines and query commands. You can also configure and monitor the U2 Web Service Server.

Organization of This Manual

This manual contains the following:

Chapter 1, "Installing U2 Web Services Development Tool," describes the step-by-step instructions to install U2 Web Services Development.

Chapter 2, "Adding and Connecting to Servers," describes how to create UniVerse and SOAP servers, and how to connect to those servers.

Chapter 3, "Creating a Query Web Service," describes how to create a web service using UniVerse RetrieVe or UniVerse SQL .

Chapter 4, "Creating a Subroutine Web Service," describes how to create a web service using a UniVerse BASIC subroutine.

Chapter 5, "Miscellaneous Features," describes miscellaneous features of the IBM U2 Web Services Developer.

Chapter 6, "Accessing the Web Services Programatically," describes how to access Web Services programatically.

Chapter 7, "Deploying Web Services," describes the U2 Web Services Developer deployment feature, which enables you to select a SOAP server defined locally and generate a deployment package, in the form of a zip file.

Documentation Conventions

This manual uses the following conventions:

Convention	Usage
Bold	In syntax, bold indicates commands, function names, and options. In text, bold indicates keys to press, function names, menu selections, and MS-DOS commands.
UPPERCASE	In syntax, uppercase indicates UniVerse commands, keywords, and options; UniVerse BASIC statements and functions; and SQL statements and keywords. In text, uppercase also indicates UniVerse identifiers such as file names, account names, schema names, and Windows file names and paths.
Italic	In syntax, italic indicates information that you supply. In text, italic also indicates UNIX commands and options, file names, and paths.
`Courier`	Courier indicates examples of source code and system output.
`Courier Bold`	In examples, courier bold indicates characters that the user types or keys the user presses (for example, **`<Return>`**).
[ ]	Brackets enclose optional items. Do not type the brackets unless indicated.
{ }	Braces enclose nonoptional items from which you must select at least one. Do not type the braces.
itemA | itemB	A vertical bar separating items indicates that you can choose only one item. Do not type the vertical bar.
...	Three periods indicate that more of the same type of item can optionally follow.
ä	A right arrow between menu options indicates you should choose each option in sequence. For example, "Choose **File** ä **Exit**" means you should choose **File** from the menu bar, then choose **Exit** from the File pull-down menu.
`I`	Item mark. For example, the item mark (`I`) in the following string delimits elements 1 and 2, and elements 3 and 4: `1I2F3I4V5`

Documentation Conventions

Convention	Usage
F	Field mark. For example, the field mark (**F**) in the following string delimits elements FLD1 and VAL1: `FLD1FVAL1VSUBV1SSUBV2`
V	Value mark. For example, the value mark (**V**) in the following string delimits elements VAL1 and SUBV1: `FLD1FVAL1VSUBV1SSUBV2`
S	Subvalue mark. For example, the subvalue mark (**S**) in the following string delimits elements SUBV1 and SUBV2: `FLD1FVAL1VSUBV1SSUBV2`
T	Text mark. For example, the text mark (**T**) in the following string delimits elements 4 and 5: `1F2S3V4T5`

Documentation Conventions (Continued)

The following conventions are also used:

- Syntax definitions and examples are indented for ease in reading.
- All punctuation marks included in the syntax—for example, commas, parentheses, or quotation marks—are required unless otherwise indicated.
- Syntax lines that do not fit on one line in this manual are continued on subsequent lines. When entering syntax, type the entire syntax entry, including the continuation lines, on the same input line.

UniVerse Documentation

UniVerse documentation includes the following:

UniVerse Installation Guide: Contains instructions for installing UniVerse 10.3.

UniVerse New Features Version 10.3: Describes enhancements and changes made in the UniVerse 10.3 release for all UniVerse products.

UniVerse BASIC: Contains comprehensive information about the UniVerse BASIC language. It is for experienced programmers.

UniVerse BASIC Commands Reference: Provides syntax, descriptions, and examples of all UniVerse BASIC commands and functions.

UniVerse BASIC Extensions: Describes the following extensions to UniVerse BASIC: UniVerse BASIC Socket API, Using CallHTTP, and Using WebSphere MQ with UniVerse.

UniVerse BASIC SQL Client Interface Guide: Describes how to use the BASIC SQL Client Interface (BCI), an interface to UniVerse and non-UniVerse databases from UniVerse BASIC. The BASIC SQL Client Interface uses ODBC-like function calls to execute SQL statements on local or remote database servers such as UniVerse, DB2, SYBASE, or INFORMIX. This book is for experienced SQL programmers.

Administering UniVerse: Describes tasks performed by UniVerse administrators, such as starting up and shutting down the system, system configuration and maintenance, system security, maintaining and transferring UniVerse accounts, maintaining peripherals, backing up and restoring files, and managing file and record locks, and network services. This book includes descriptions of how to use the UniAdmin program on a Windows client and how to use shell commands on UNIX systems to administer UniVerse.

Using UniAdmin: Describes the UniAdmin tool, which enables you to configure UniVerse, configure and manage servers and databases, and monitor UniVerse performance and locks.

UniVerse Security Features: Describes security features in UniVerse, including configuring SSL through UniAdmin, using SSL with the CallHttp and Socket interfaces, using SSL with UniObjects for Java, and automatic data encryption.

UniVerse Transaction Logging and Recovery: Describes the UniVerse transaction logging subsystem, including both transaction and warmstart logging and recovery. This book is for system administrators.

UniVerse System Description: Provides detailed and advanced information about UniVerse features and capabilities for experienced users. This book describes how to use UniVerse commands, work in a UniVerse environment, create a UniVerse database, and maintain UniVerse files.

UniVerse User Reference: Contains reference pages for all UniVerse commands, keywords, and user records, allowing experienced users to refer to syntax details quickly.

Guide to RetrieVe: Describes RetrieVe, the UniVerse query language that lets users select, sort, process, and display data in UniVerse files. This book is for users who are familiar with UniVerse.

Guide to ProVerb: Describes ProVerb, a UniVerse processor used by application developers to execute prestored procedures called procs. This book describes tasks such as relational data testing, arithmetic processing, and transfers to subroutines. It also includes reference pages for all ProVerb commands.

Guide to the UniVerse Editor: Describes in detail how to use the Editor, allowing users to modify UniVerse files or programs. This book also includes reference pages for all UniVerse Editor commands.

UniVerse NLS Guide: Describes how to use and manage UniVerse's National Language Support (NLS). This book is for users, programmers, and administrators.

UniVerse SQL Administration for DBAs: Describes administrative tasks typically performed by DBAs, such as maintaining database integrity and security, and creating and modifying databases. This book is for database administrators (DBAs) who are familiar with UniVerse.

UniVerse SQL User Guide: Describes how to use SQL functionality in UniVerse applications. This book is for application developers who are familiar with UniVerse.

UniVerse SQL Reference: Contains reference pages for all SQL statements and keywords, allowing experienced SQL users to refer to syntax details quickly. It includes the complete UniVerse SQL grammar in Backus Naur Form (BNF).

Related Documentation

The following documentation is also available:

UniVerse GCI Guide: Describes how to use the General Calling Interface (GCI) to call subroutines written in C, C++, or FORTRAN from UniVerse BASIC programs. This book is for experienced programmers who are familiar with UniVerse.

UniVerse ODBC Guide: Describes how to install and configure a UniVerse ODBC server on a UniVerse host system. It also describes how to use UniVerse ODBC Config and how to install, configure, and use UniVerse ODBC drivers on client systems. This book is for experienced UniVerse developers who are familiar with SQL and ODBC.

UV/Net II Guide: Describes UV/Net II, the UniVerse transparent database networking facility that lets users access UniVerse files on remote systems. This book is for experienced UniVerse administrators.

UniVerse Guide for Pick Users: Describes UniVerse for new UniVerse users familiar with Pick-based systems.

Moving to UniVerse from PI/open: Describes how to prepare the PI/open environment before converting PI/open applications to run under UniVerse. This book includes step-by-step procedures for converting INFO/BASIC programs, accounts, and files. This book is for experienced PI/open users and does not assume detailed knowledge of UniVerse.

API Documentation

The following books document application programming interfaces (APIs) used for developing client applications that connect to UniVerse and UniData servers.

Administrative Supplement for Client APIs: Introduces IBM's seven common APIs, and provides important information that developers using any of the common APIs will need. It includes information about the UniRPC, the UCI Config Editor, the *ud_database* file, and device licensing.

UCI Developer's Guide: Describes how to use UCI (Uni Call Interface), an interface to UniVerse and UniData databases from C-based client programs. UCI uses ODBC-like function calls to execute SQL statements on local or remote UniVerse and UniData servers. This book is for experienced SQL programmers.

IBM JDBC Driver for UniData and UniVerse: Describes UniJDBC, an interface to UniData and UniVerse databases from JDBC applications. This book is for experienced programmers and application developers who are familiar with UniData and UniVerse, Java, JDBC, and who want to write JDBC applications that access these databases.

InterCall Developer's Guide: Describes how to use the InterCall API to access data on UniVerse and UniData systems from external programs. This book is for experienced programmers who are familiar with UniVerse or UniData.

UniObjects Developer's Guide: Describes UniObjects, an interface to UniVerse and UniData systems from Visual Basic. This book is for experienced programmers and application developers who are familiar with UniVerse or UniData, and with Visual Basic, and who want to write Visual Basic programs that access these databases.

UniObjects for Java Developer's Guide: Describes UniObjects for Java, an interface to UniVerse and UniData systems from Java. This book is for experienced programmers and application developers who are familiar with UniVerse or UniData, and with Java, and who want to write Java programs that access these databases.

UniObjects for .NET Developer's Guide: Describes UniObjects, an interface to UniVerse and UniData systems from .NET. This book is for experienced programmers and application developers who are familiar with UniVerse or UniData, and with .NET, and who want to write .NET programs that access these databases.

Using UniOLEDB: Describes how to use UniOLEDB, an interface to UniVerse and UniData systems for OLE DB consumers. This book is for experienced programmers and application developers who are familiar with UniVerse or UniData, and with OLE DB, and who want to write OLE DB programs that access these databases.

Chapter

1

Installing U2 Web Services Development Tool

The chapter describes how to install the IBM U2 Web Services Developer.

To access the IBM U2 Web Services Developer, you must have UniVerse 10.1.18 or greater, with connection pooling licensed.

Installation Process

1. Log On As an Administrator

You must log on with Administrator privileges to install IBM U2 Web Service Developer. Either log on to the Administrator account on the Windows system, or log on as a member of the local Administrators group.

2. Exit Other Applications

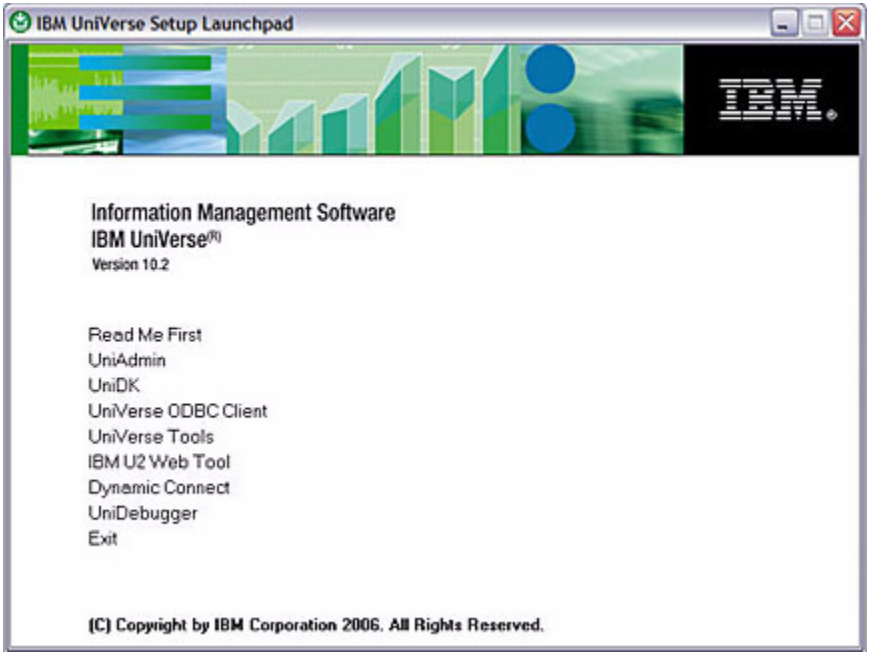
Before proceeding, exit any other Windows applications you may have open.

3. Load the UniVerse Client CD

Place the UniVerse Client CD in your CD-ROM drive. Make sure you have the UniVerse Product Configuration sheet that is included with your media. You will need this form when you license the product after installation.

4. Execute the Installation

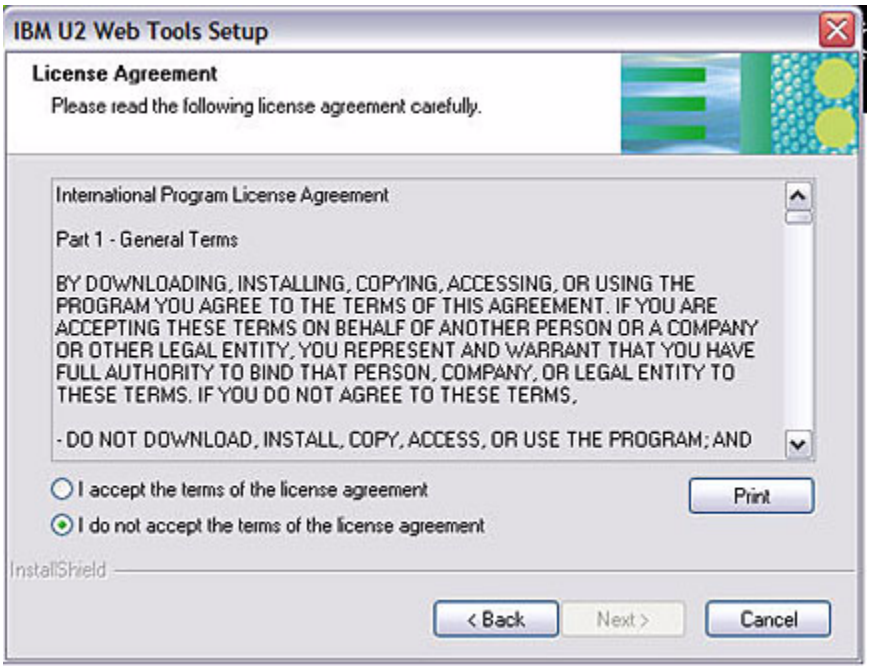
From the **Setup** screen, click **IBM U2 Web Tool**, as shown in the following example:



The **Welcome** window appears. Click **Next**.

5. Review License Agreement

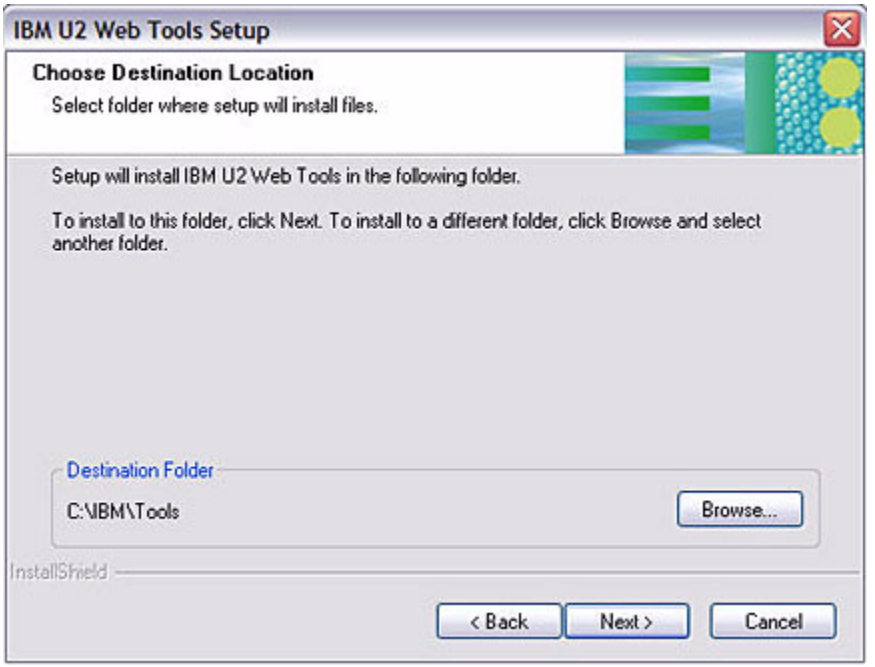
The **License Agreement** dialog box appears, as shown in the following example:



Review the license agreement. If you agree with the terms, select **I accept the terms of the license agreement**. If you do not agree, select **I do not accept the terms of the license agreement**. If you agree with the terms of the license agreement, click **Next**. to proceed with the installation.

6. Choose Installation Location

The **Choose Destination Location** dialog box appears, as shown in the following example:

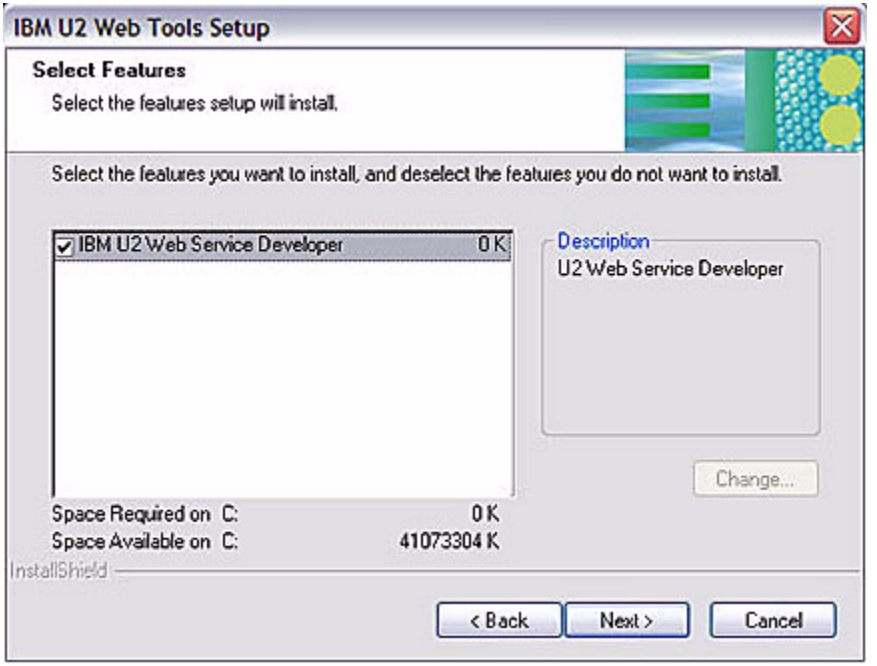


By default, U2 Web Services Developer is installed in the C:\IBM\Tools folder. If you want to install U2 Web Services Developer in a different location, click **Browse** to select that location.

Click **Next** to proceed with the installation.

7. Select Features to Install

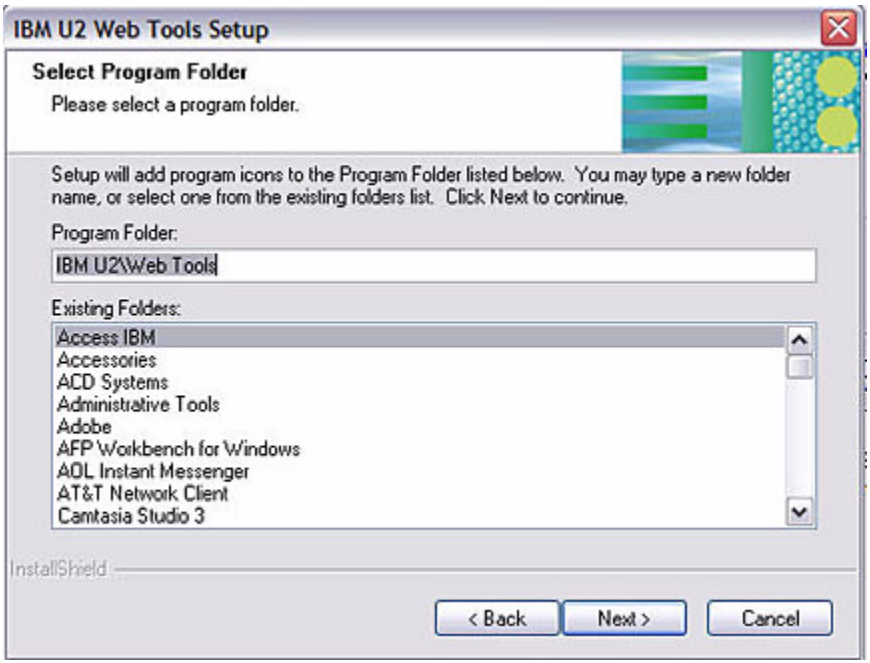
Select the features you want to install, as shown in the following example:



At Universe 10.3, the only web tool available to install is IBM U2 Web Services Developer. Click **Next** to proceed with the installation.

8. Select Program Folder

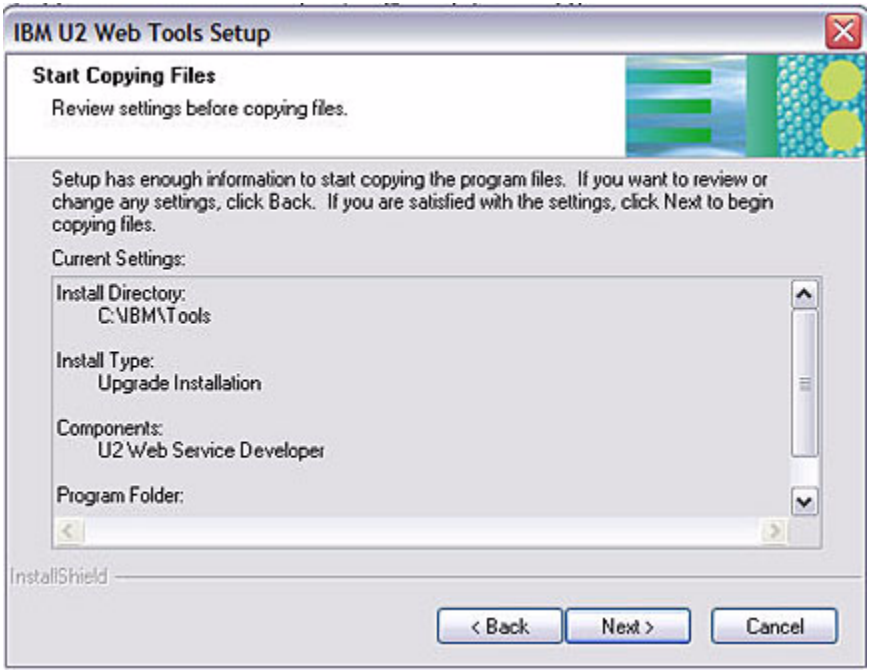
The **Select Program Folder** dialog box appears, as shown in the following example:



By default, IBM U2 Web Services Developer is installed in the IBM U2\Web Tools folder. If you do not want to install it in this folder, select the folder where you want to install the product and click **Next** to continue with the installation.

9. Copy Files

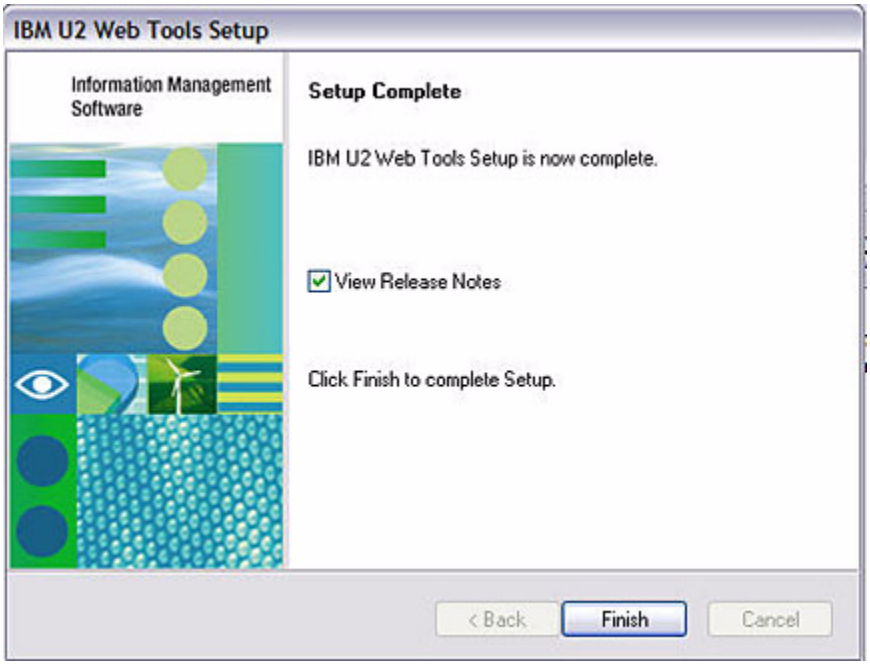
At this point, the installation process has enough information to begin copying files. Review the information in the **Start Copying Files** dialog box, as shown in the following example:I



If all of the information is correct, click **Next** to proceed with the installation. If it is not correct, click **Back** and correct the appropriate information.

10. Complete Installation

When the installation is complete, the **Setup Complete** dialog box appears, as shown in the following example:

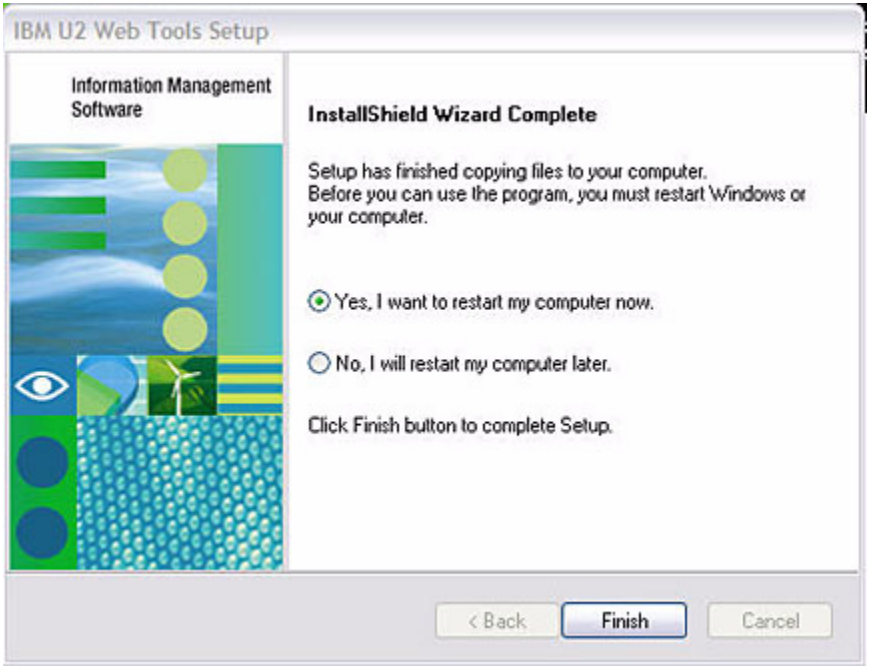


If you want to review the release notes, select the **View Release Notes** box.

Click **Finish** to complete the installation.

11. Restart Your Computer

Before you can use IBM U2 Web Services Developer, you must restart your computer. Choose to restart your computer now or at a later time in the following dialog box:



Click **Finish** to complete the installation.

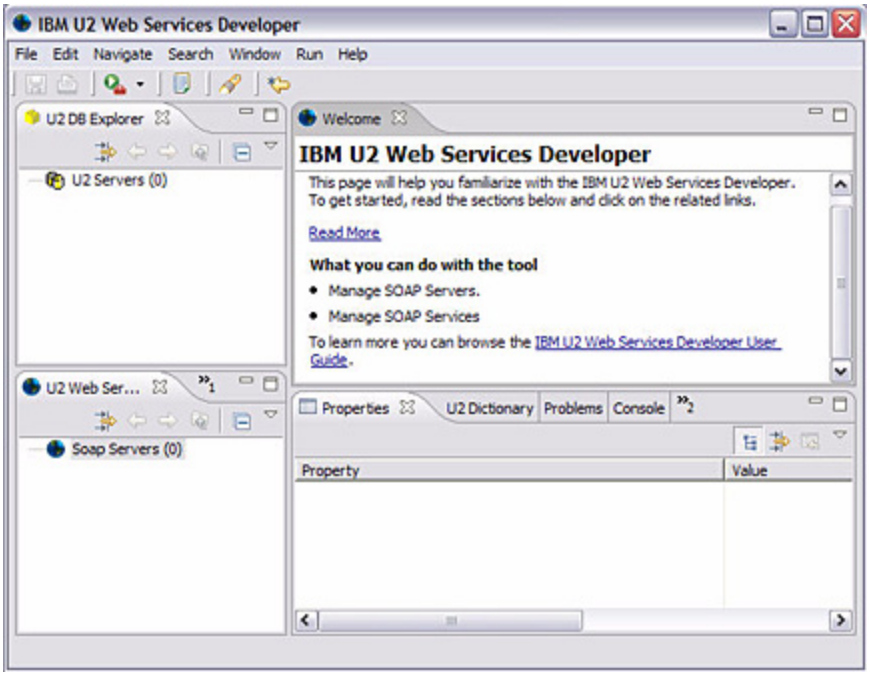
Chapter

2

Adding and Connecting to Servers

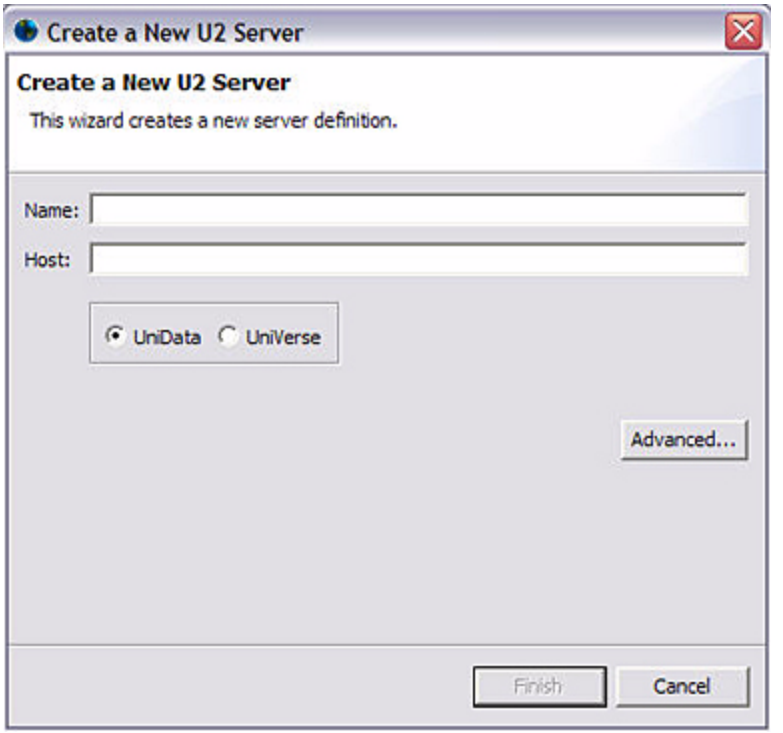
The chapter describes how to create UniVerse and SOAP servers, and how to connect to those servers.

From the **Start** menu, select **All Programs**, select **IBM U2**, select **Web Tools**, then click **IBM U2 Web Services Developer**. A window similar to the following example appears:



Add a UniData or UniVerse Server

From the **IBM U2 Web Services Developer** window, with the right mouse button (right-click), click **U2 Servers**, then click **New U2 Server**. The **Create New U2 Server** dialog appears, as shown in the following example:

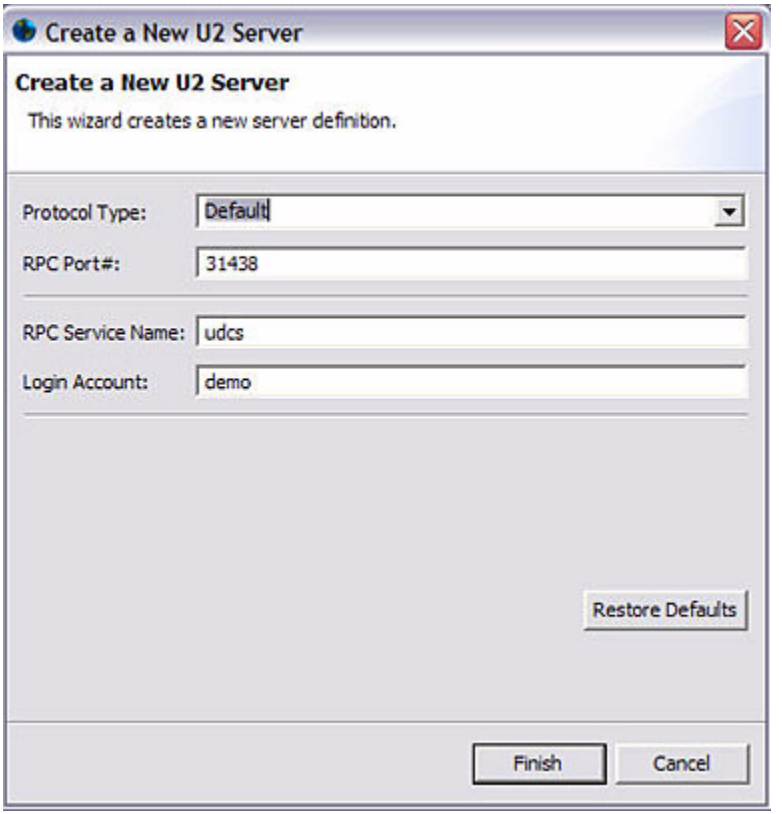


In the **Name** box, enter a unique name of the UniVerse or UniData server.

In the **Host** box, enter a valid network name for the host.

Select either UniData or UniVerse to define the database you are using.

If you want to define the Protocol Type, RPC port number, RPC Service Name, or the account to access, click **Advanced**. The following screen appears:



Protocol Type

In the **Protocol Type** box, make sure the type of communication you are using to the server is Default or TCP/IP.

RPC Port Number

In the **RPC Port #** box, enter the port number of the UniRPC server running on the host. The default port number is 31438.

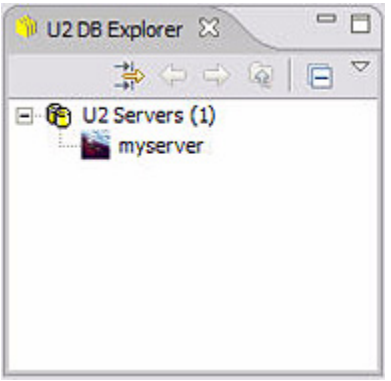
RPC Service Name

In the **RPC Service Name** box, enter the name of the RPC service on your system. For UniVerse, this is normally uvcs.

Login Account

In the **Login Account** box, enter the name of the account to which you want to log on when accessing UniVerse..

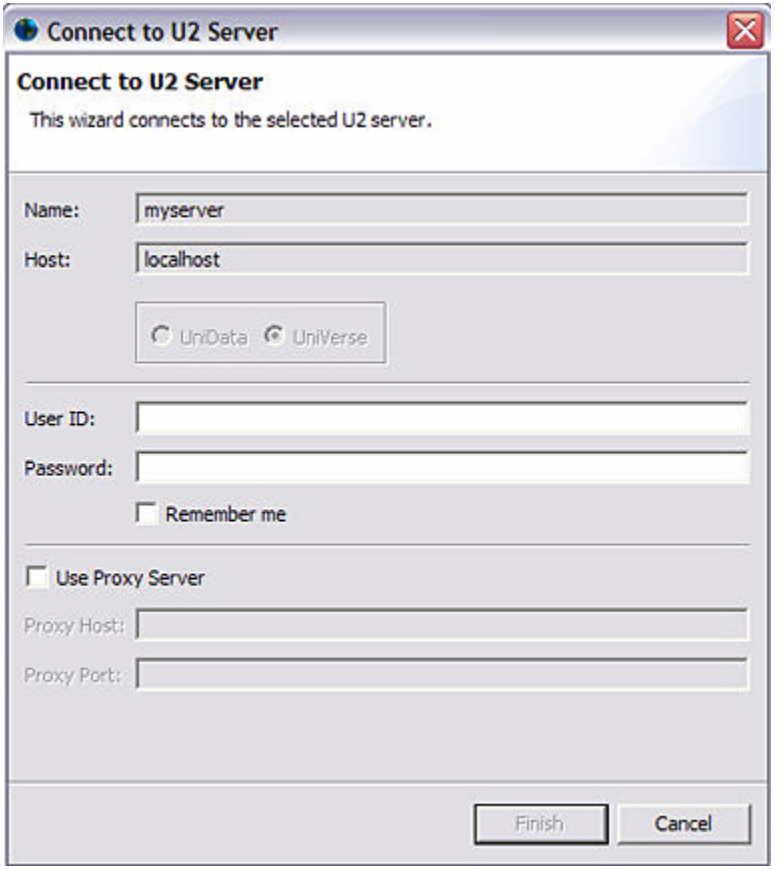
Click **Finish**. The server you added appears in the **U2 DB Explorer** portion of the **IBM U2 Web Services Developer** window under **U2 Servers**, as shown in the following example:



Connect to the UniData or UniVerse Database

Make sure that UniVerse or UniData are running, and the unirpc daemon is started on the remote host.

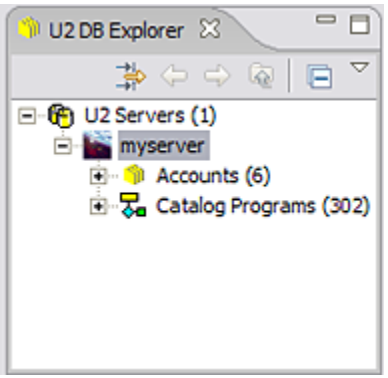
Double-click the server to which you want to connect to the database, then click **Connect**. The **Connect to U2 Server** dialog box appears, as shown in the following example:



In the **User ID** box, enter your log on name for the server. Enter the corresponding password in the **Password** box.

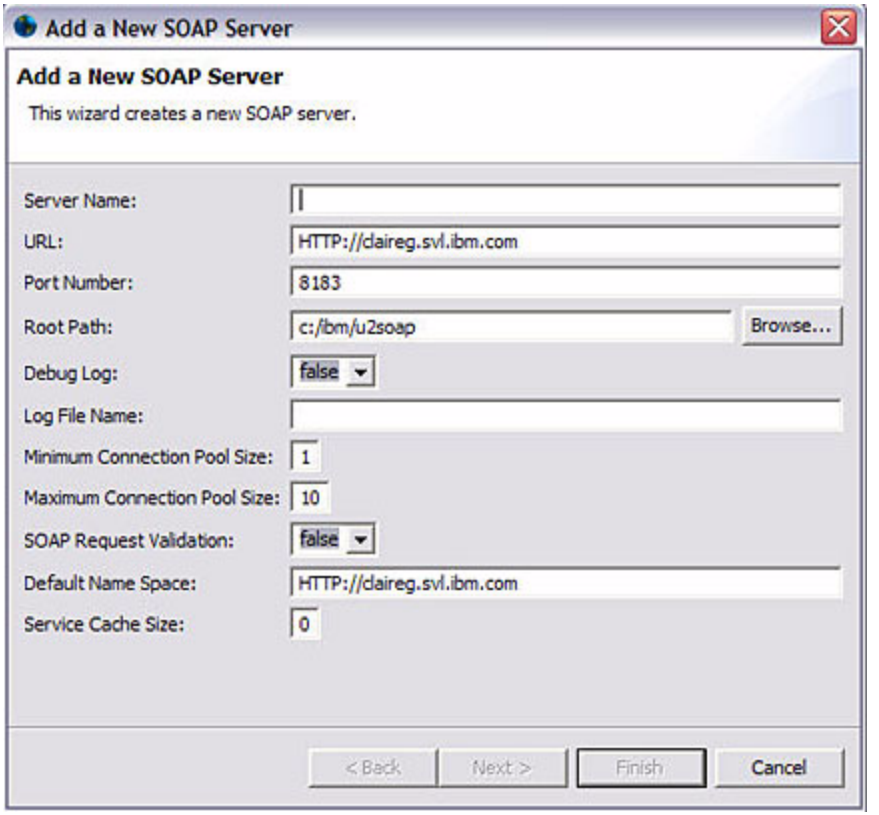
If you do not want to enter your user ID and password in subsequent sessions, select the **Remember Me** check box.

Click **Finish**. The existing accounts and globally cataloged programs appear under the UniVerse or UniData server name, as shown in the following example:



Create SOAP Server

In the **U2 Web Services** portion of the **IBM U2 Web Services Developer** window, right-click **Soap Server**, then click **New SOAP Server.** The **Add a New SOAP Server** dialog box appears, as shown in the following example:



In the **Server Name** box, enter a unique name for the SOAP server.

The remainder of the fields are populated based on information retrieved by the IBM U2 Web Services Developer tool. You can change any of these fields.

URL

The URL for the SOAP server you specify. The URL is automatically detected by the IBM U2 Web Services Developer. We recommend that you not change the URL unless you are sure the new URL you specify is valid and accessible.

Port Number

The port number on which the server will listen.

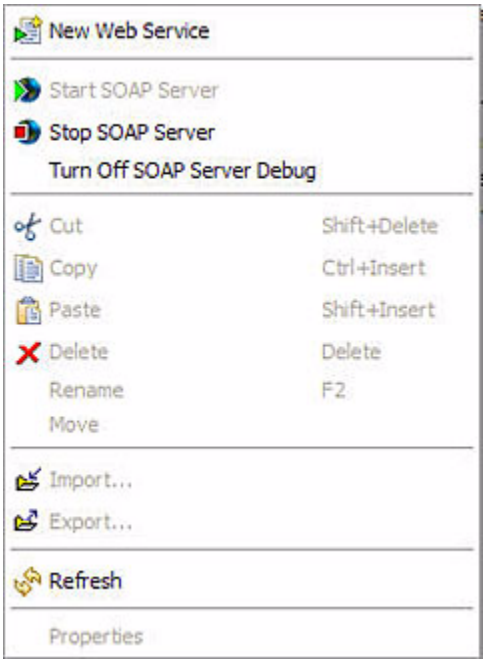
***Note**: IBM recommends that you not change the port number unless that port number is used by another service.*

Root Path

The path to the root directory where the definitions to the web services are stored.

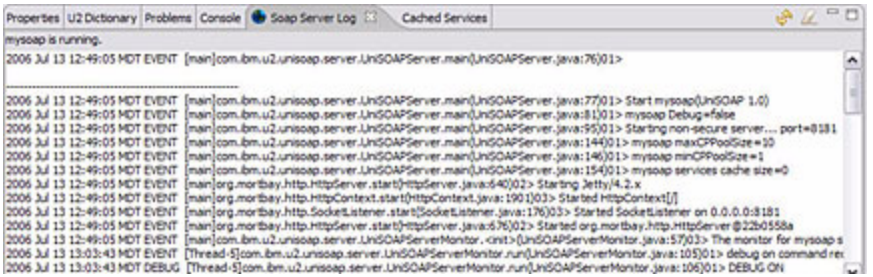
Debug Log

Indicates whether to start the debug log. If you select **true**, U2 IBM Web Services Developer starts the debug log each time you start the server. To disable the debug log when the server is running, right-click the SOAP server, then select **Turn Off SOAP Server Debug**, as shown in the following example:



You can also start the debug log when the SOAP server is running, if it is disabled.

Click **Soap Server Log** to display the SOAP Server Logs, as shown in the following example:



IBM U2 Web Services developer displays the last 64KB in the SOAP Server log.

To erase the contents of the log file click the eraser icon, as shown in the following example:

To refresh the contents of the log file, click the circular arrows icon, as shown in the following example:

Log File Name

The name of the debug log file. By default, the name of the debug log file is *soapservername*.log.

Connection Pooling On

The **Connection Pooling On** setting determines if UniVerse uses connection pooling with the U2 Web Services Developer. The term *connection pooling* refers to the technology that pools permanent connections to data sources for multiple threads to share. It improves application performance by saving the overhead of making a fresh connection each time one is required. Instead of physically terminating a connection when it is no longer needed, connections are returned to the pool and an available connection is given to the next thread with the same credentials.

Connection Pooling is enabled by default. If you do not want to use Connection Pooling, clear the **Connection Pooling On** check box.

***Note**: IBM recommends using Connection Pooling for superior performance and readability.*

Connection Pool Size

You can set the minimum and maximum size of the connection pool. If you do not define these sizes, the minimum size defaults to 1 and the maximum size defaults to 10. The minimum size determines the initial size of the connection pool.

The size of the connection pool changes dynamically between the minimum and maximum sizes you specify, depending on the system demands. When there are no pooled connections available, UniVerse either creates another connection, if the maximum connection pool size has not been reached, or keeps the thread waiting in the queue until a pooled connection is released or the request times out. If a pooled connection is idle for a specified time, it is disconnected.

SOAP Request Validation

Specifies whether the server needs to validate the SOAP request before processing. If this value is set to **true**, you may experience slight performance degradation, but will have an extra layer of security.

Default Name Space

The name space for the Web Services you define.

Service Cache Size

For performance purposes, you can set this value to a number greater than 0 to indicate the number of web service definitions you want to keep in the cache. If you set this value, the SOAP Server will always try to read the web service definition from the cache first. If you do not set this value, the SOAP Server reloads the web service each time from disk.

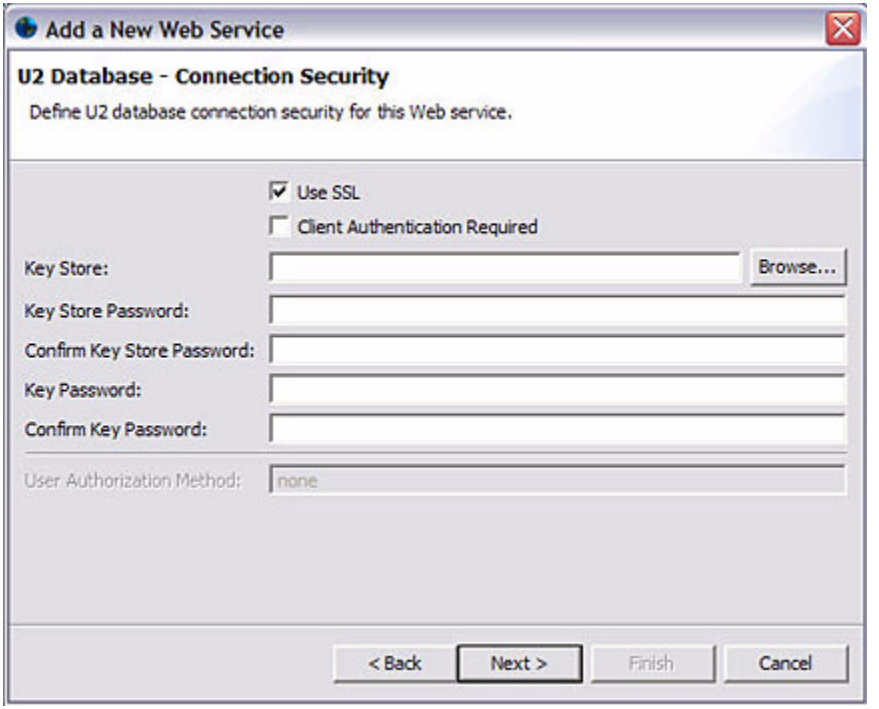
If you are developing a web service, we recommend keeping this value at 0. This setting forces the SOAP Server to reload the new web service definition each time.

Select the **Cached Services** tab to display the web services currently loaded in cache.

Click **Next**.

Define SSL Settings

If you want to use SSL with the SOAP Server, select the **Use SSL** check box. The **SOAP Server - Connection Security** dialog box appears, as shown in the following example:



Key Store

Enter the full path to the key store on the SOAP server, or click **Browse** to navigate to the key store.

Key Store Password

Enter the password corresponding to the key store you defined in the **Key Store** box.

In the **Confirm Key Store Password** box, reenter the password.

Key Password

Enter the encryption key password, if one exists, in the **Key Password** box. Reenter the password in the **Confirm Key Password** box.

Enable Authentication

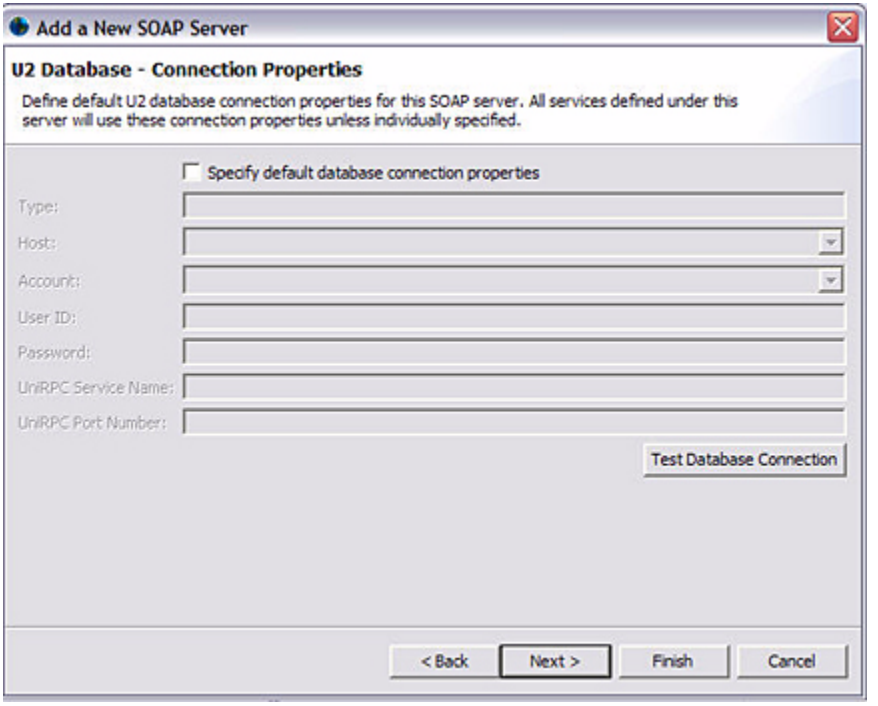
If you want the client to send its certification for authentication, select the **Need Client Authentication** check box.

Click **Next**.

For detailed information about SSL, see *UniVerse Security Features*.

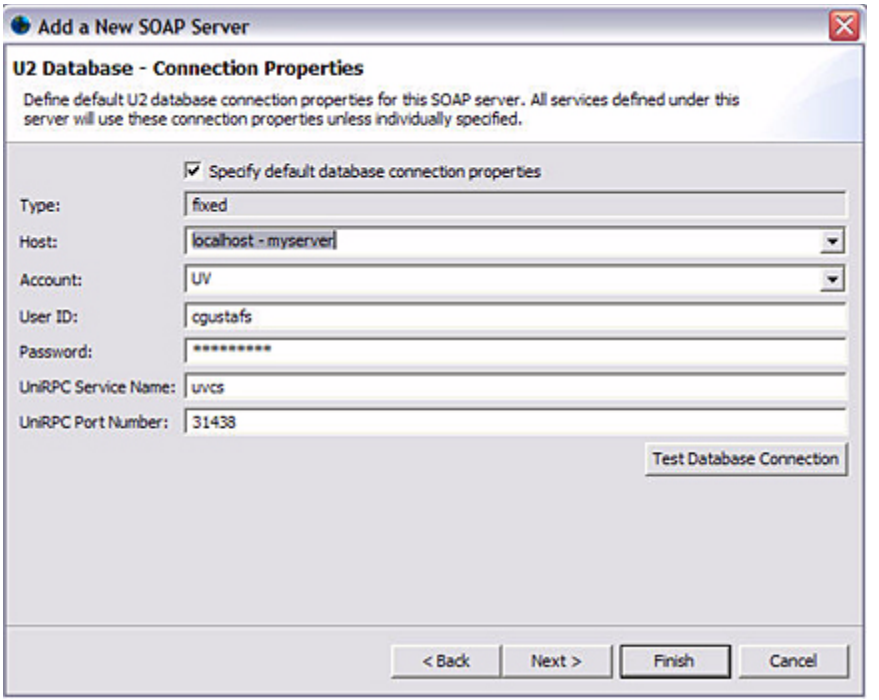
Set Connection Properties

The **U2 Database - Connection Properties** dialog box appears, as shown in the following example:



Select the **Specify default database connection properties** check box.

In the **Host** box, select host name from the list of available UniData or UniVerse servers. The remaining fields in the dialog box are populated based on the information retrieved by the IBM U2 Web Services Developer, as shown in the following example:



To test the connection to the database, click **Test Database Connection**.

The following message appears when the connection to the UniData or UniVerse server is successful.

From the **U2 Database - Connection Properties** dialog box, click **Finish**.

Chapter

3

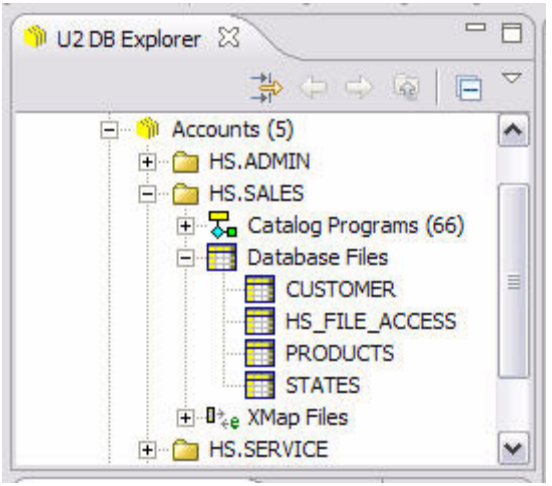
Creating a Query Web Service

This chapter describes how to create a web service using UniVerse RetrieVe or UniVerse SQL.

Note: You cannot publish a query that requires interactive user input as a web service.

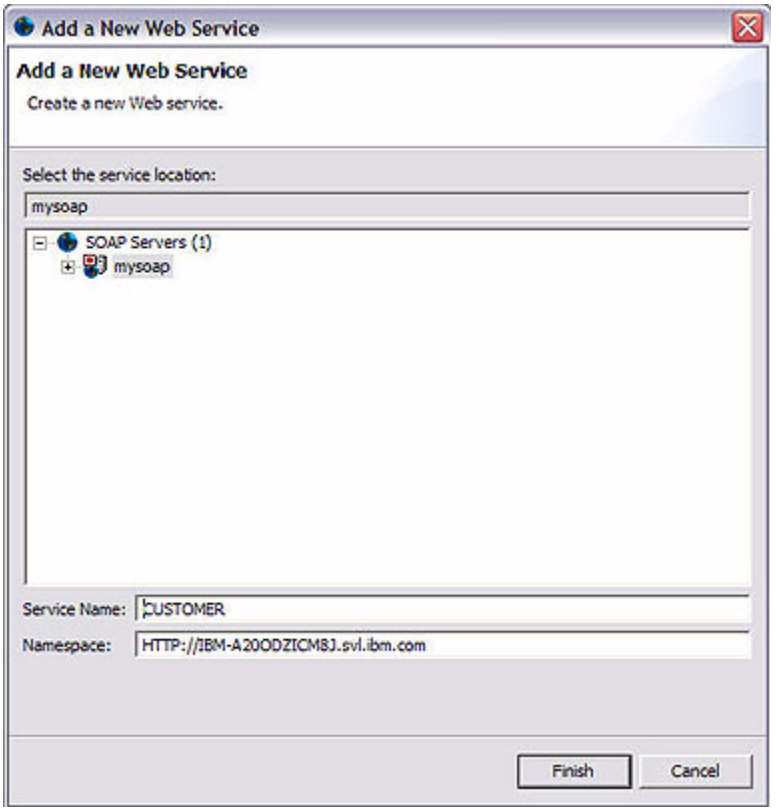
Creating a Query Web Service Using a Drag-and-Drop Operation

From the **IBM U2 Web Services Developer** window, right-click the account for which you want to create a web service. Click the plus sign ("+") next to **Database Files** to view the database files available in the account, as shown in the following example:



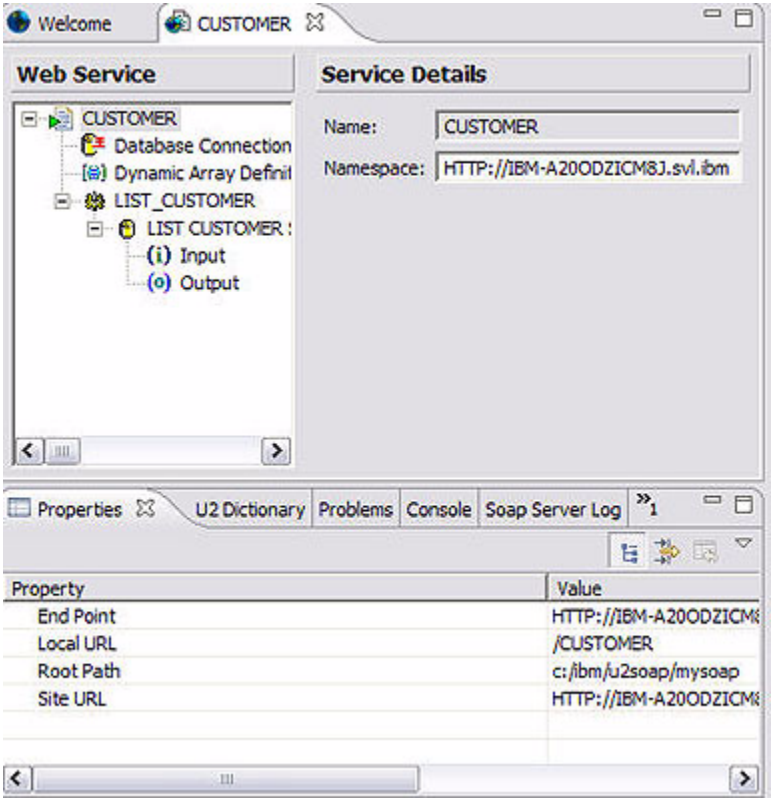
Using a drag-and-drop operation, move the file for which you want to create a web service to the appropriate SOAP Server.

The **Add a New Web Service** dialog box appears, as shown in the following example:



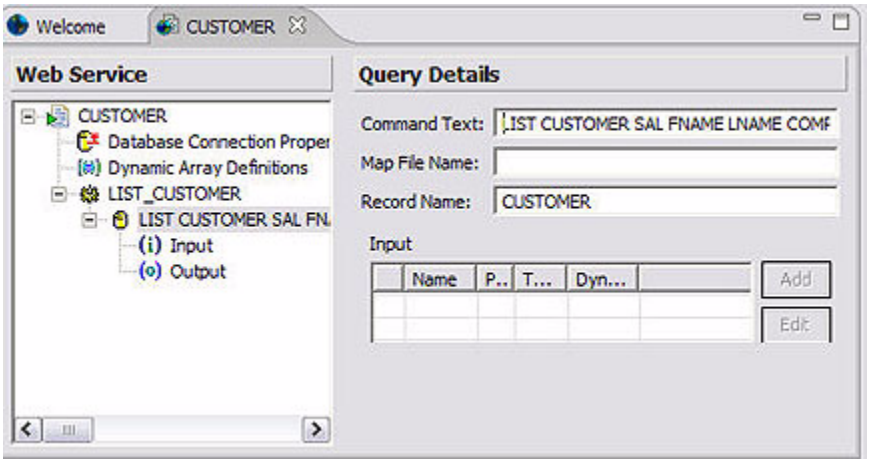
Enter a name for the web service you are creating in the **Service Name** box.

Verify or enter the namespace in the **Namespace** box. Click **Finish**. Information about the SOAP service appears in the **Web Service** portion of the **IBM U2 Web Services Developer** dialog box, as shown in the following example:



By default, the IBM U2 Web Services Developer creates a LIST statement which includes each D-Type dictionary record contained in dictionary for the file you selected.

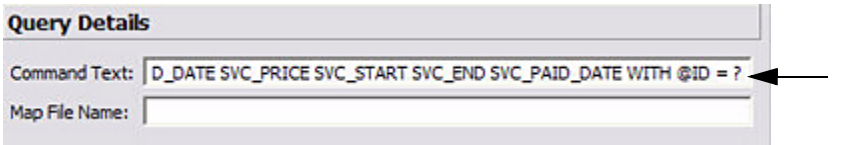
Click the query statement to view the details for the statement. The **Query Details** dialog box appears, as shown in the following example:



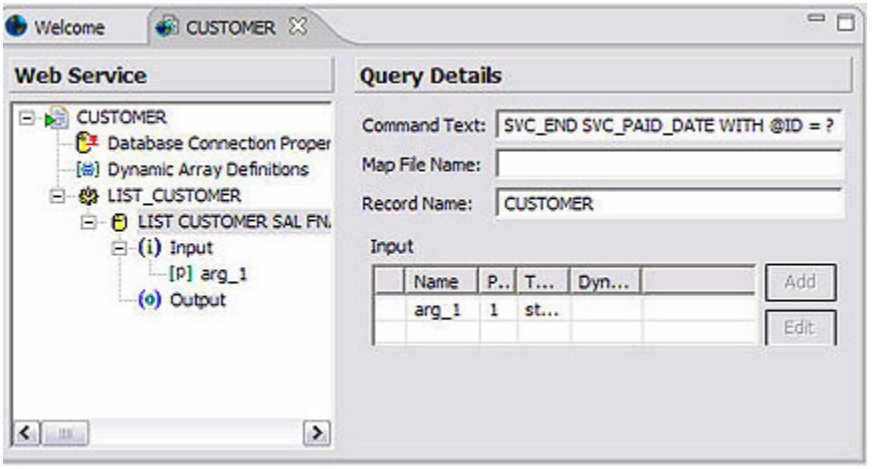
UniData and UniVerse only support the UniQuery or UniVerse RetrieVe LIST and SORT commands, or the SQL SELECT command.

Adding an Input Parameter

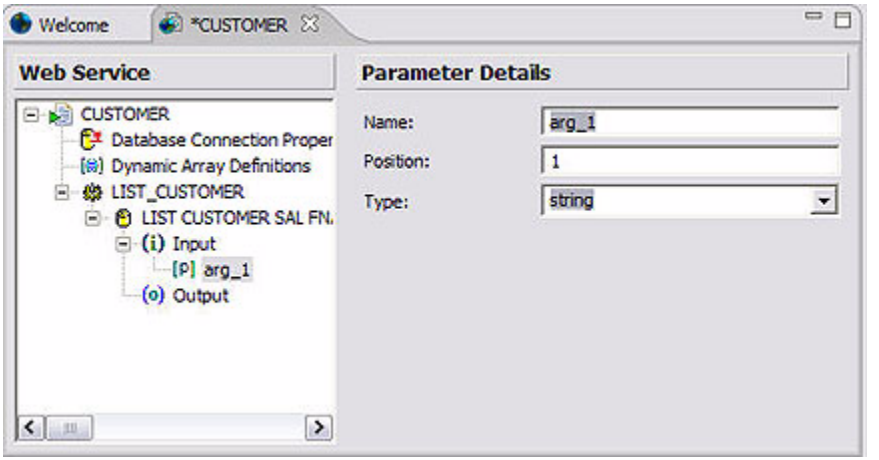
If you want to add an input parameter, modify the statement that appears in the **Command Text** box, as shown in the following example:



The question mark (“?”) acts as a placeholder for the input value. Press CTRL+S to save the query, or click the **Save** icon. The input parameter now appears under the Web Service Input parameter, as shown in the following example:



Click the input parameter. The **Parameter Details** dialog box appears, as shown in the following example:



In the **Name** box, enter a meaningful name of the input parameter. This is the name for which you are prompted when you invoke the service.

In the **Position** box, enter the prompting order of the input parameter.

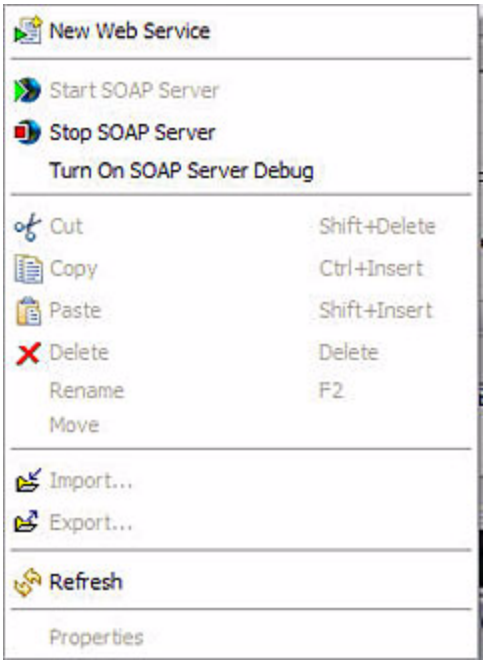
In the **Type** box, select the data type for the input parameter from the list. Valid types are:

- String
- Dynamic array

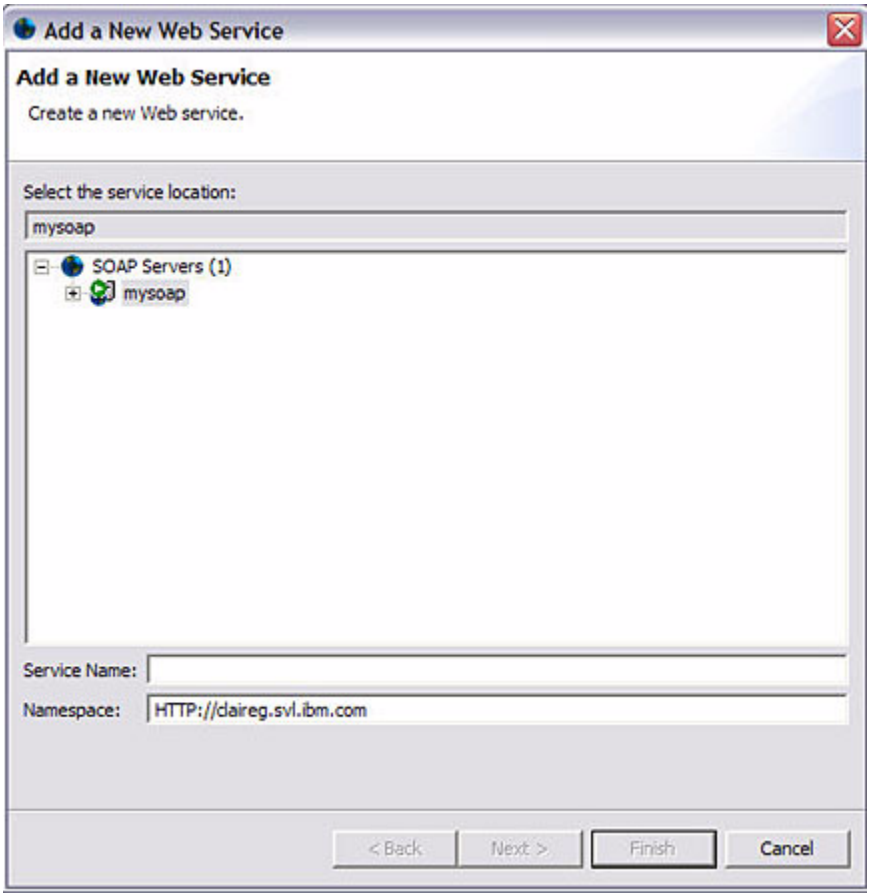
Enter CTRL+S or the **Save** icon to save your changes.

Creating a Query Web Service Using the Wizard

To create a Query Web Service using the wizard, right-click the SOAP server where you want to create the service, then click **Create New Web Service**, as shown in the following example:



The **Add a New Web Service** dialog box appears, as shown in the following example:

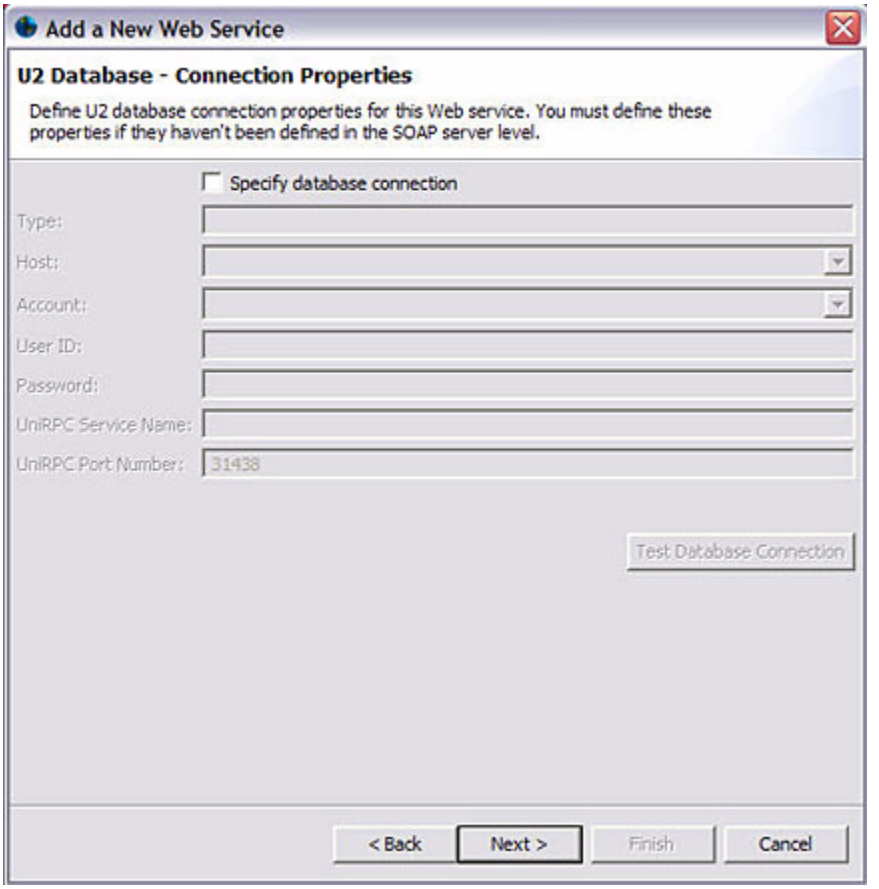


Enter the name of the service in the **Service Name** box.

Verify or enter the namespace for the service in the **Namespace** box.

Click **Next**.

The **U2 Database - Connection Properties** dialog box appears, as shown in the following example:



If you have not defined the database connection properties at the SOAP Server-level, you must define them for this Web Service.

Select the **Specify database connection** check box.

Type

The type of database connection. At this release, the only type of connection supported is “fixed.”

Host

Select the name of the host server from the list of UniVerse servers you have defined. The server should be running.

Account

Select the account name on the UniVerse server you specified where you want attach when you connect from the list. This account must contain the data files you are accessing with the query.

User ID and Password

In the **User ID** box, enter your log on name for the server. Enter the corresponding password in the **Password** box.

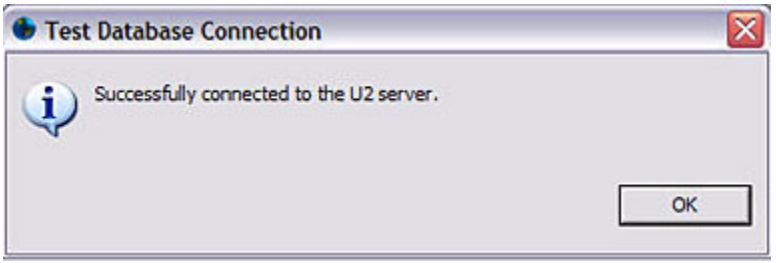
UniRPC Service Name

Enter the appropriate UniRPC Service Name in the UniRPC **Service Name** box. For UniVerse, the service name is uvcs. For UniData, the service name is udcs.

UniRPC Port Number

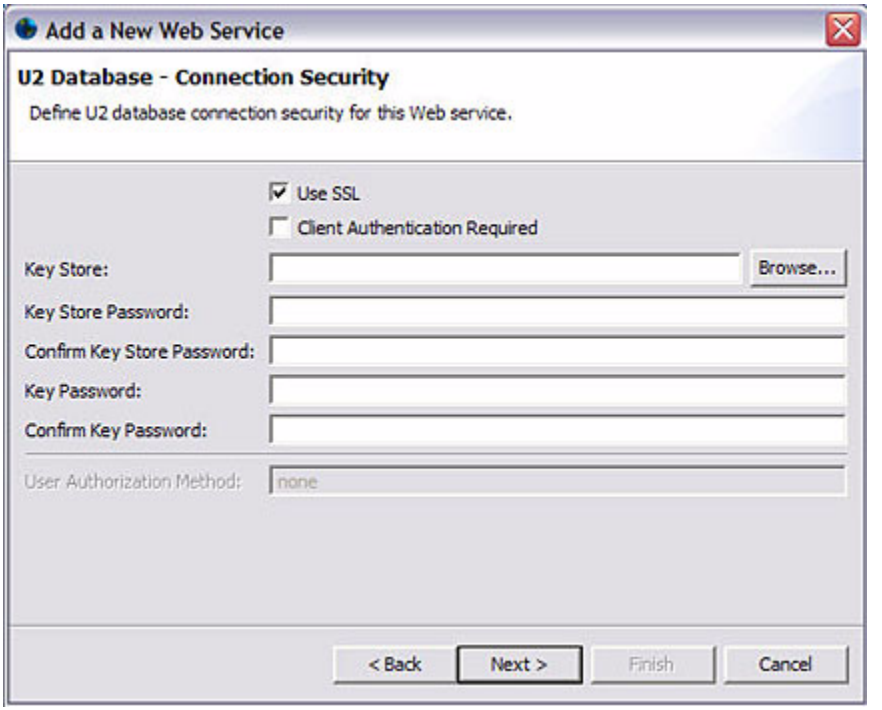
Enter the port number of the UniRPC server running on the host in the **UniRPC Port Number** box. The default port number is 31438.

To test the connection to the database, click **Test Database Connection**. If the connection is successful, the following message appears:



Click **Next.**

The **U2 Database - Connection Security** dialog box appears, as shown in the following example:



Key Store

Enter the full path to the key store on the SOAP server, or click **Browse** to navigate to the key store.

Key Store Password

Enter the password corresponding to the key store you defined in the **Key Store** box.

In the **Confirm Key Store Password** box, reenter the password.

Key Password

Enter the encryption key password, if one exists, in the **Key Password** box. Reenter the password in the **Confirm Key Password** box.

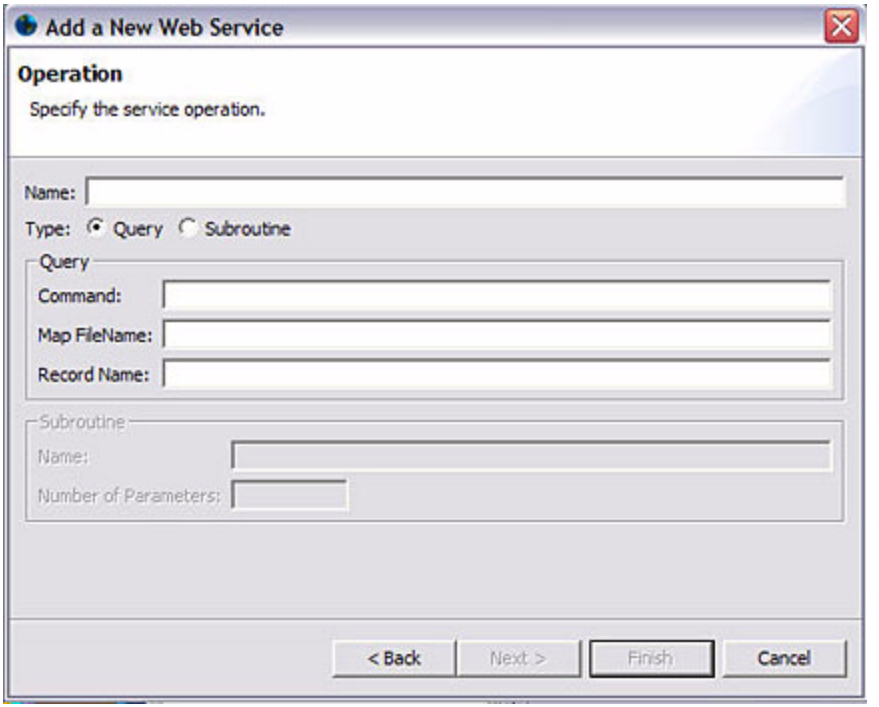
Enable Authentication

If you want the client to send its certification for authentication, select the **Client Authentication Required** check box.

For detailed information about SSL, see *UniVerse Security Features*.

Click **Next**.

The **Operation** dialog box appears, as shown in the following example:t



Name

Enter the name for the Web Service you are creating in the **Name** box.

Type

Select **Query** as the type of Web Service you are creating.

Define Query

In the **Query** area of the **Operation** dialog box, define the Query Command, Map File Name, and Record Name for the Web Service.

Query Command

Enter the RetrieVe or UniVerse SQL statement in the **Command** box. At this release U2 IBM Web Services Developer only support the UniVerse RetrieVe LIST and SORT commands, or the UniVerse SQL SELECT statement.

If you want to add an input parameter, use a question mark (“?”) as a placeholder for the input value. Press CTRL+S to save the query, or click the **Save** icon. The input parameter appears under the Web Service Input parameter

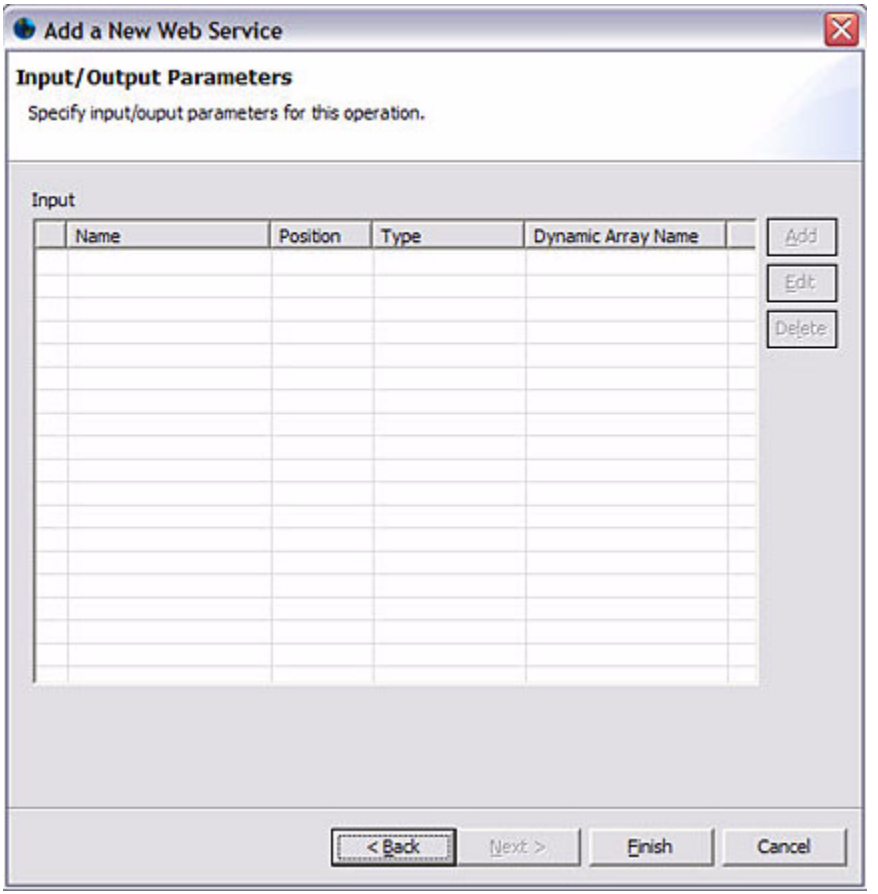
Map File Name

The name of the mapping file you want to use, located in the &XML& directory. This field is optional. See *UniVerse Guide to RetrieVe* for detailed information about the mapping file.

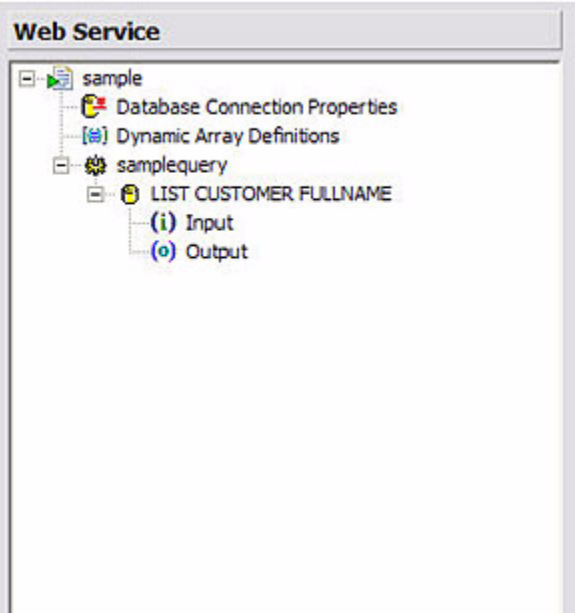
Record Name

The name of the root element in the XML document the web service creates for the query you specify. If you have two or more query operations within the same web service using the same data file, make sure the record name is different for each operation.

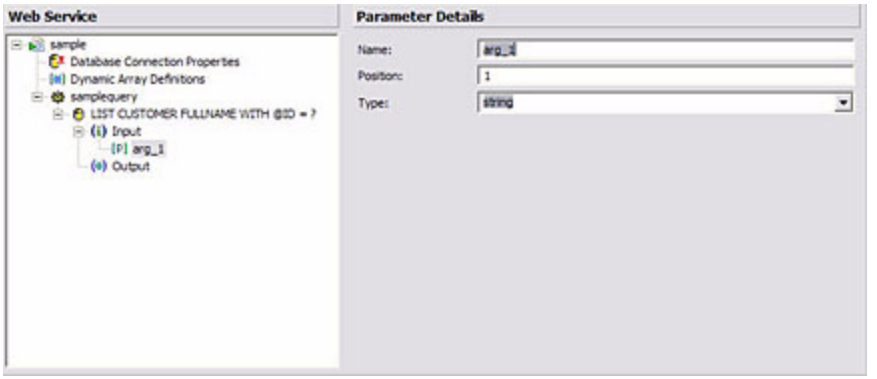
Click **Next**. The **Input/Output Parameters** dialog box appears, as shown in the following example.



Click **Finish**. The web service definition appears in the **Web Service** area of IBM U2 Web Services Developer window, as shown in the following example:



If you specified a placeholder in the web service command, click the input parameter, then click arg_1. The **Parameter Details** dialog box appears, as shown in the following example:



In the **Name** box, enter a meaningful name of the input parameter. This is the name for which you are prompted when you invoke the service.

In the **Position** box, enter the prompting order of the input parameter.

In the **Type** box, select the data type for the input parameter from the list. Valid types are:

- String
- Dynamic array

Enter CTRL+S or the **Save** icon to save your changes.

Executing the Web Service

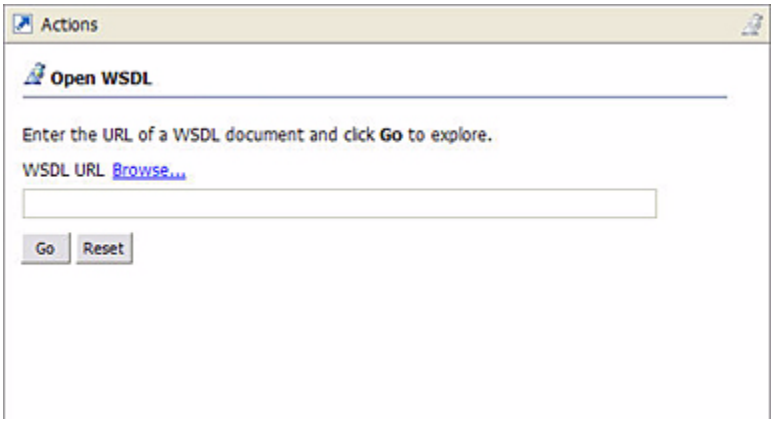
To execute a Web Service, click the **Launch** icon from the toolbar:

Launch icon

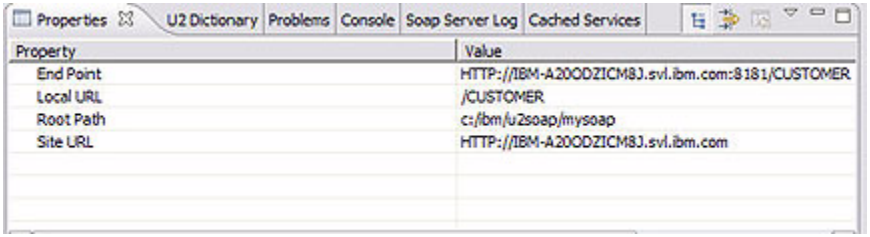
The **Web Services Explorer** window appears. Click the **WSDL** icon, as shown in the following example:

WSDL icon

Under the **Navigator** area of the **Web Services Explorer** window, click **WSDL Main**. The **Open WSDL** dialog box appears, as shown in the following example:

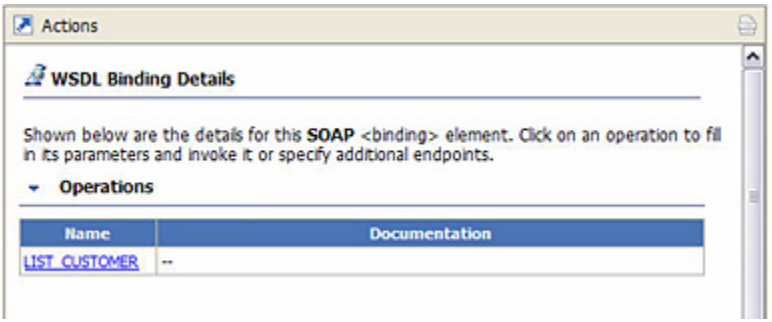


From the **U2 Web Service** area of the **IBM U2 Web Services Developer** window, click the highlighted web service. The properites of the web service appear in the **Properties** area of the **Web Service** window, as shown in the following example:

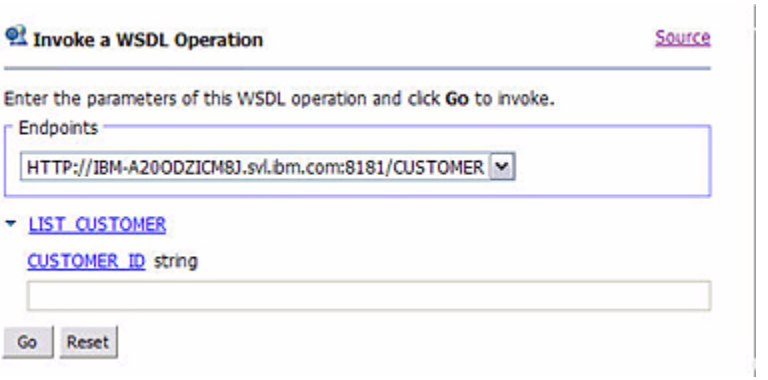


In the WSDL URL dialog box, enter the value of the **End Point** property, then click **Go**.

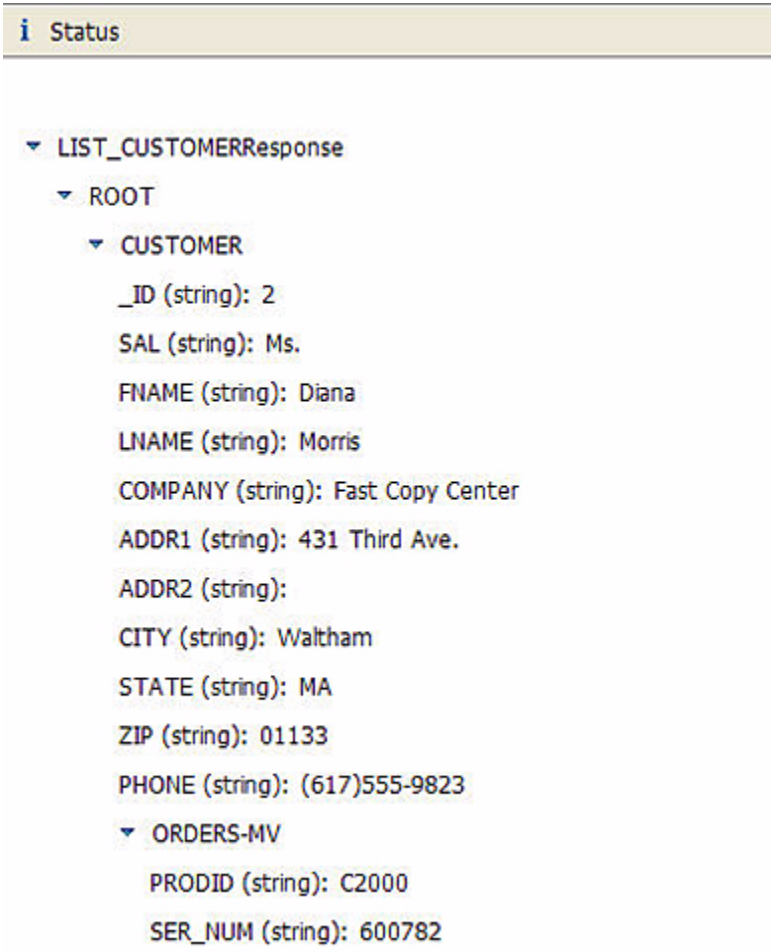
In the **WSDL Details** window, click the operation you want to execute, as shown in the following example:



The Web Service operation prompts you for input parameters you previously defined, as shown in the following example:



Enter the input value, then click **GO**. The IBM U2 Web Services Developer displays the resulting XML document in the **Status** area of the **Invoke a WSDL Operation** dialog box, as shown in the following example:



Click **Source** to view the XML source for the output.

Chapter

4

Creating a Subroutine Web Service

This chapter describes how to create a web service using a UniVerse BASIC subroutine.

Note: You cannot publish a subroutine that requires interactive user input as a web service.

Creating a Subroutine Web Service Using a Drag-and-Drop Operation

From the **IBM U2 Web Services Developer** window, right-click the account for which you want to create a web service. Click the plus sign (“+”) next to **Cataloged Program** to view the subroutines available in the account, as shown in the following example:

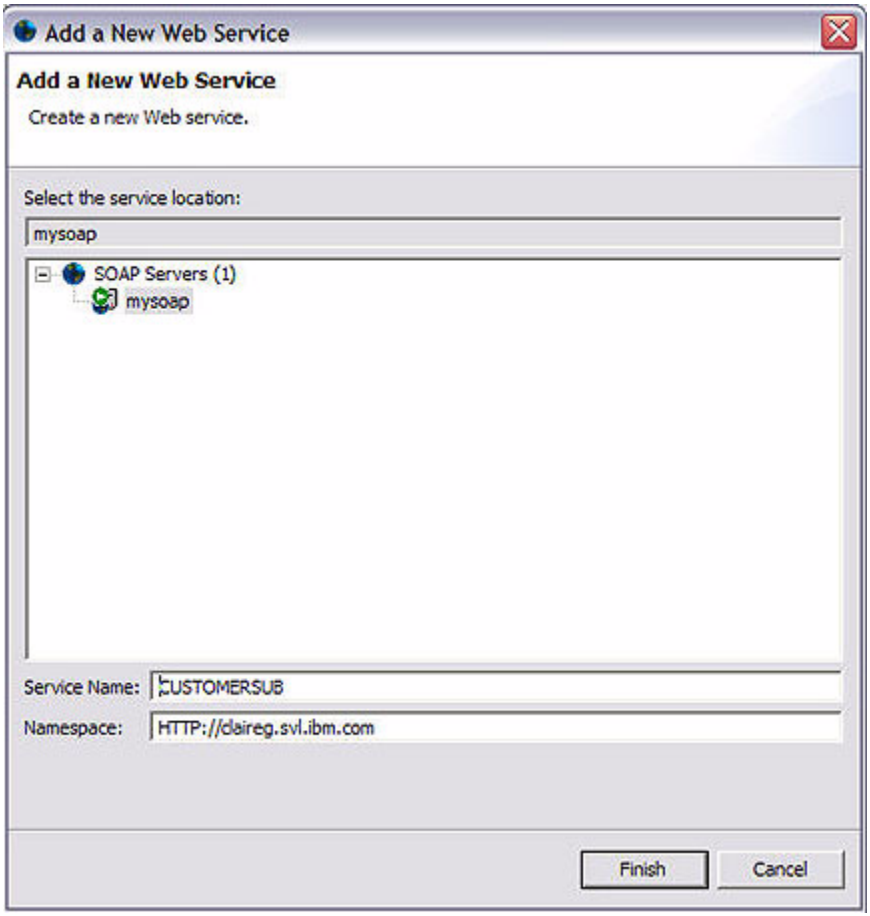
***Note**: You must catalog the subroutine before you start the IBM U2 Web Services Developer. Subroutines may be cataloged globally, locally, or directly.*

For information about cataloging UniBasic programs, see the UniData Commands Reference. For information about cataloging UniVerse BASIC programs, see the UniVerse User Reference.

In this example, we are using the following subroutine:

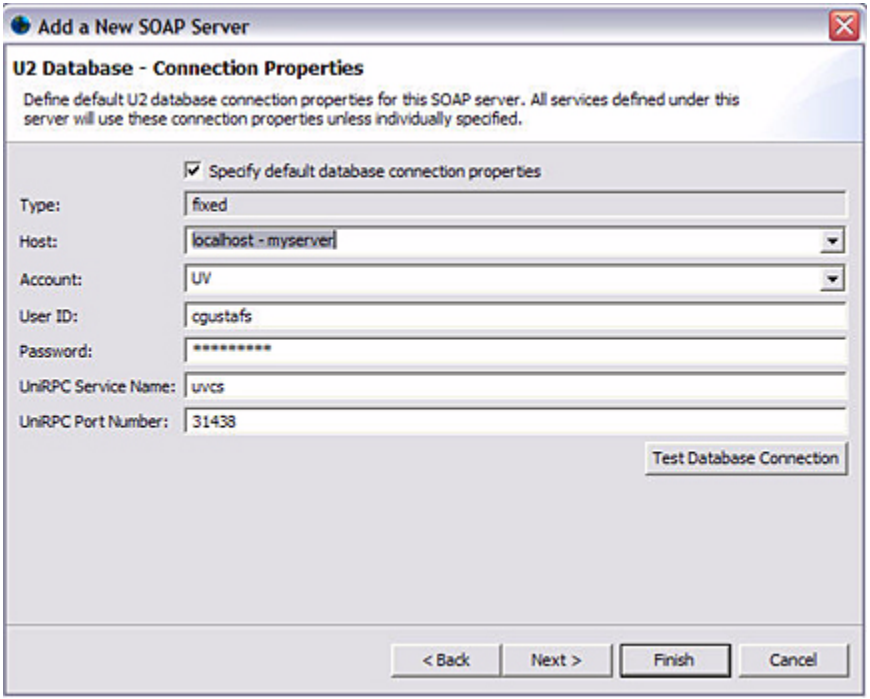
```
SUBROUTINE CUSTOMERSUB(ID, REC)
OPEN "CUSTOMER" TO F.CUST ELSE REC = ""
READ REC FROM F.CUST, ID ELSE REC = ""
CLOSE F.CUST
RETURN
```

Using a drag-and-drop operation, move the subroutine for which you want to create a web service to the appropriate SOAP Server.



Enter a name for the web service you are creating in the **Service Name** box.

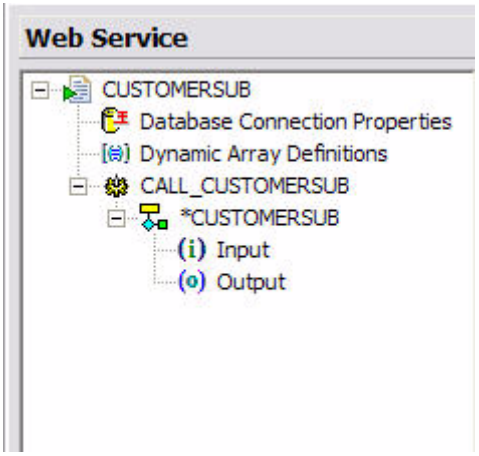
If the subroutine is globally cataloged, click **Next**. Click the **Specify default database connection properties** check box. The **U2 Database - Connection Properties** dialog box appears, as shown in the following example:



Since the program is globally cataloged, the **Connection Properties** dialog box appears. Verify the **Account** box to make sure it defines the account you want to execute the subroutine against, then click **Finish**.

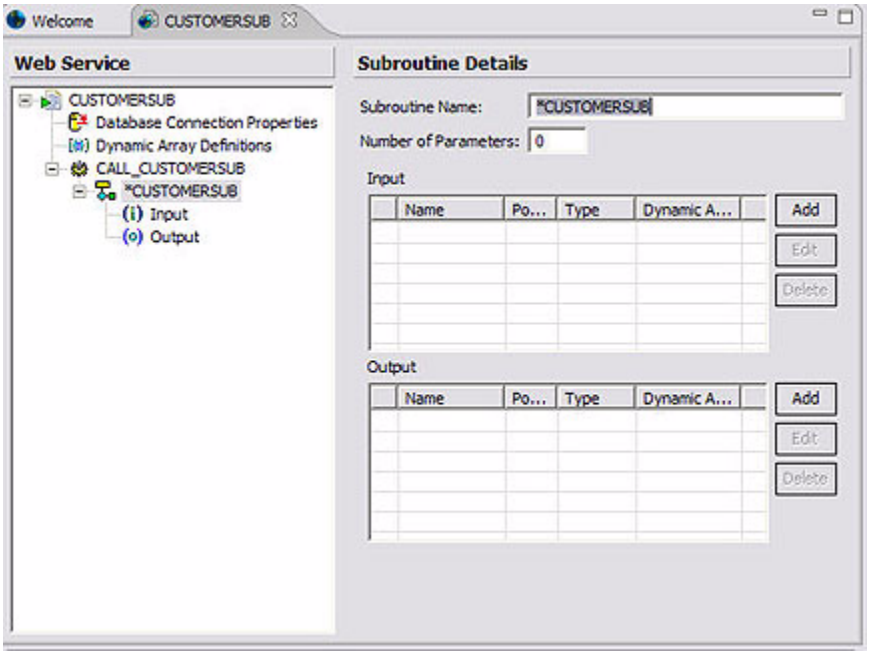
If the subroutine is not globally cataloged, the **Connection Properties** dialog box does not appear.

Verify or enter the namespace in the **Namespace** box. Click **Finish**. Information about the SOAP service appears in the **Web Service** portion of the **IBM U2 Web Services Developer** dialog box, as shown in the following example:



Define Subroutine Parameters

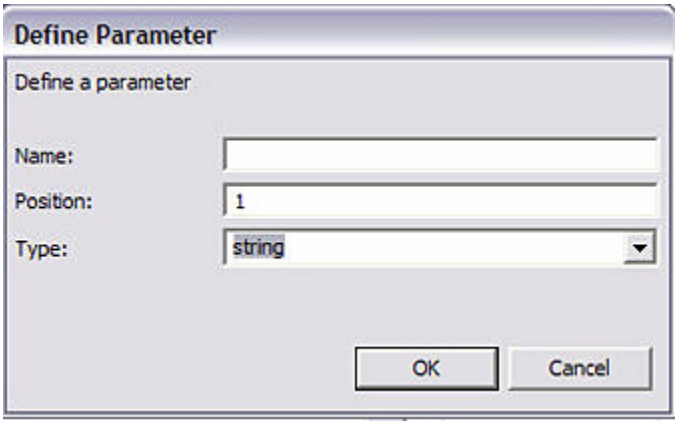
Click the subroutine for which you are defining parameters. The **Subroutine Details** dialog box appears:



Enter the name of the subroutine in the **Subroutine Name** box. Enter the total number of parameters defined for the subroutine in the **Number of Parameters** box. In the example subroutine, we have two parameters, one input, and one output.

Define Input Parameters

To define the input parameters for the subroutine, click **Add** in the **Input** area of the **Subroutine Details** dialog box. The **Define Parameter** dialog box appears:



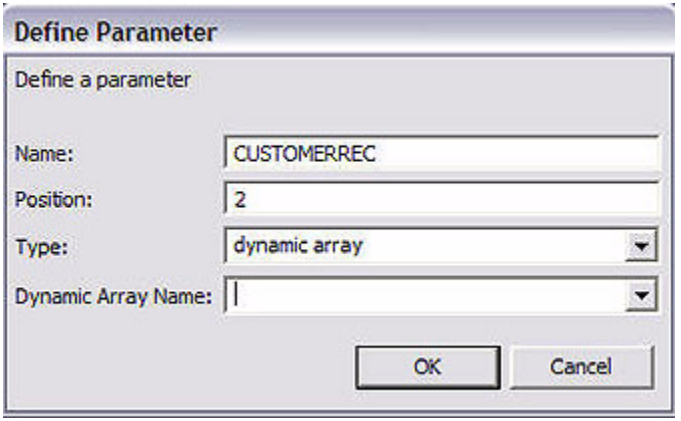
In the example subroutine, the input parameter is the Customer ID, and it is the first parameter in the subroutine.

In the **Name** box, enter a meaningful name for the parameter. This name does not have to be the same as the one defined in the UniVerse BASIC program. Enter the position the input parameter appears in the subroutine in the **Position** box. In the **Type** box, enter the data type for the input parameter, the click **OK**.

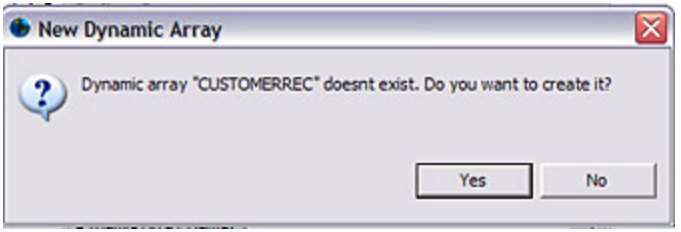
Define Output Parameters

To define the output parameters for the subroutine, click **Add** in the **Output** area of the **Subroutine Details** dialog box. The **Define Parameter** dialog box appears.

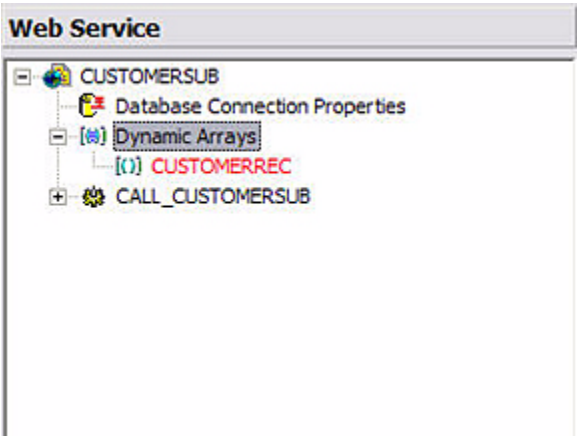
Enter the name of the output parameter in the **Name** box. Enter the position of the output parameter in the subroutine in the **Position** box. Enter the data type of the output parameter in the **Type** box. In our example, the output parameter is a dynamic array.



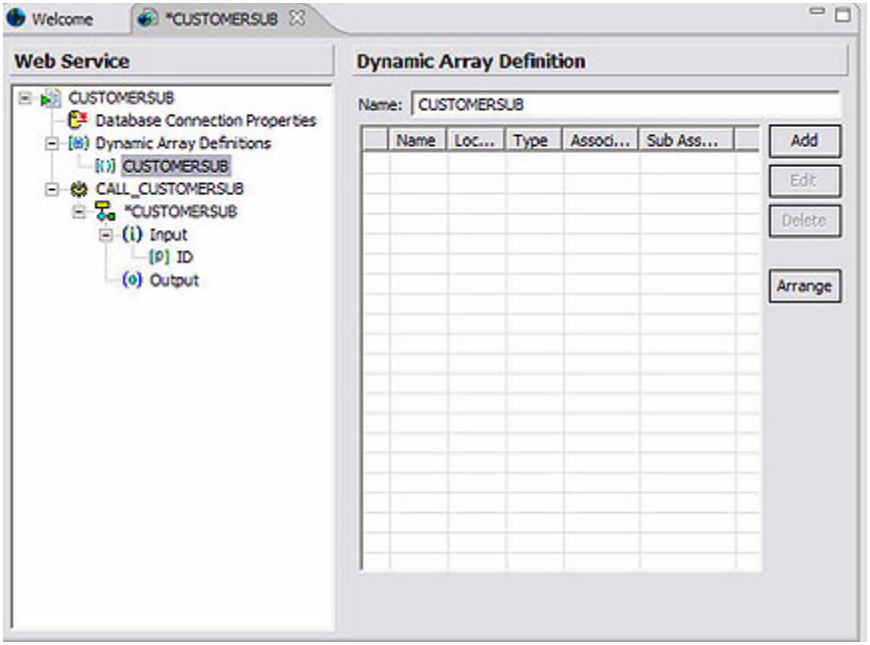
Enter the name of the dynamic array in the **Dynamic Array Name** box. You can choose an existing dynamic array, or create a new one. If the dynamic array you specify does not exist, a message similar to the following example appears:



Click **Yes** to define a new dynamic array. The name of the array appears under the **Dynamic Arrays** area of the **Web Services** window, as shown in the following example:

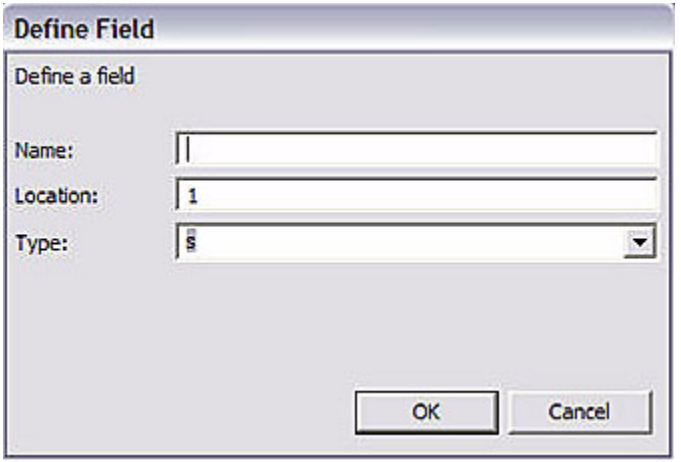


Double-click the dynamic array for which you want to define fields. The **Dynamic Array Fields** dialog box appears, as shown in the following example:



You can enter the dynamic arrays manually, or populate them using a drag-and-drop operation.

To manually add a field, click **Add**. The **Define Field** dialog box appears, as shown in the following example:



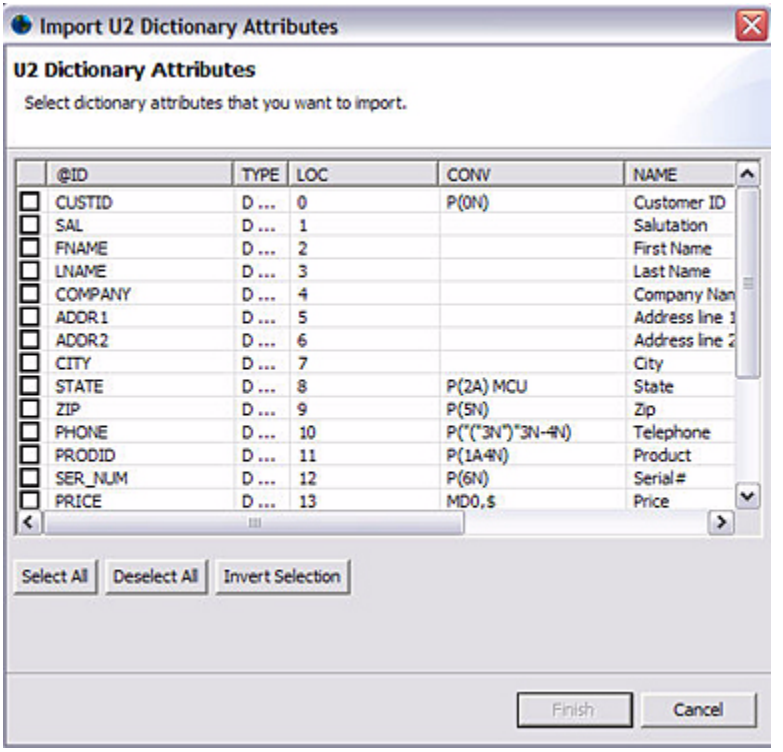
Enter the name of the field in the dynamic array in the **Name** box.

Enter the location of the field in the dynamic array in the **Location** box.

Enter the type of field in the **Type** box. Valid types are:

- s – Singlevalued
- mv – Multivalued
- ms – Multi-subvalued

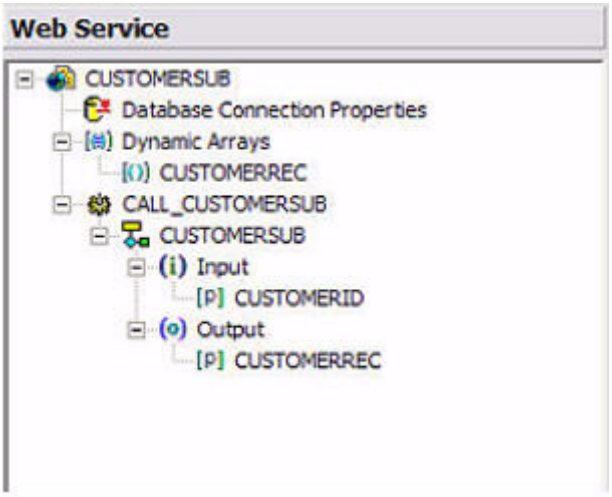
To populate the fields, from the Database Files in the account for which you are creating the web service, move the file pertaining to the dynamic array using a drag-and-drop operation. The IBM U2 Web Services Developer populates the **U2 Dictionary** dialog box with each D-type dictionary record, as shown in the following example:



Select the check boxes next to the dictionary record ID you want to include in the dynamic array. The number of attributes you select must match the number of fields in the parameter in the subroutine. In our example, we are returning the entire record, so each dictionary attribute is selected, except for the CUSTID.

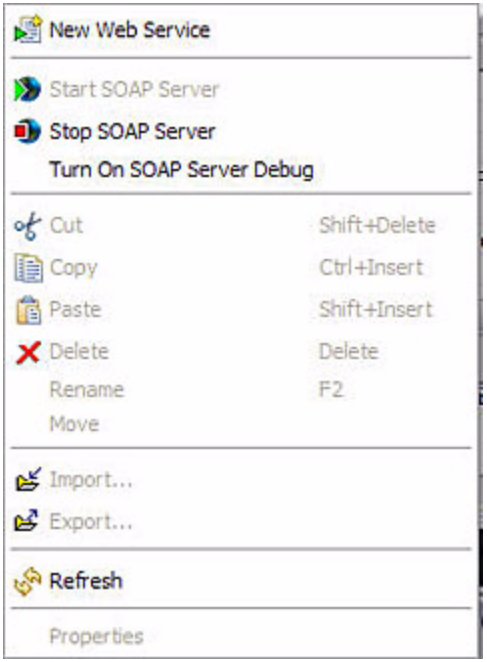
Enter CTRL+S or the **Save** icon to save your dynamic array definition.

When you have finished defining the parameters of the subroutine, the input parameters, output parameters, and dynamic arrays appear in the **Web Services** area of the **IBM U2 Web Services Developer** window, as shown in the following example:

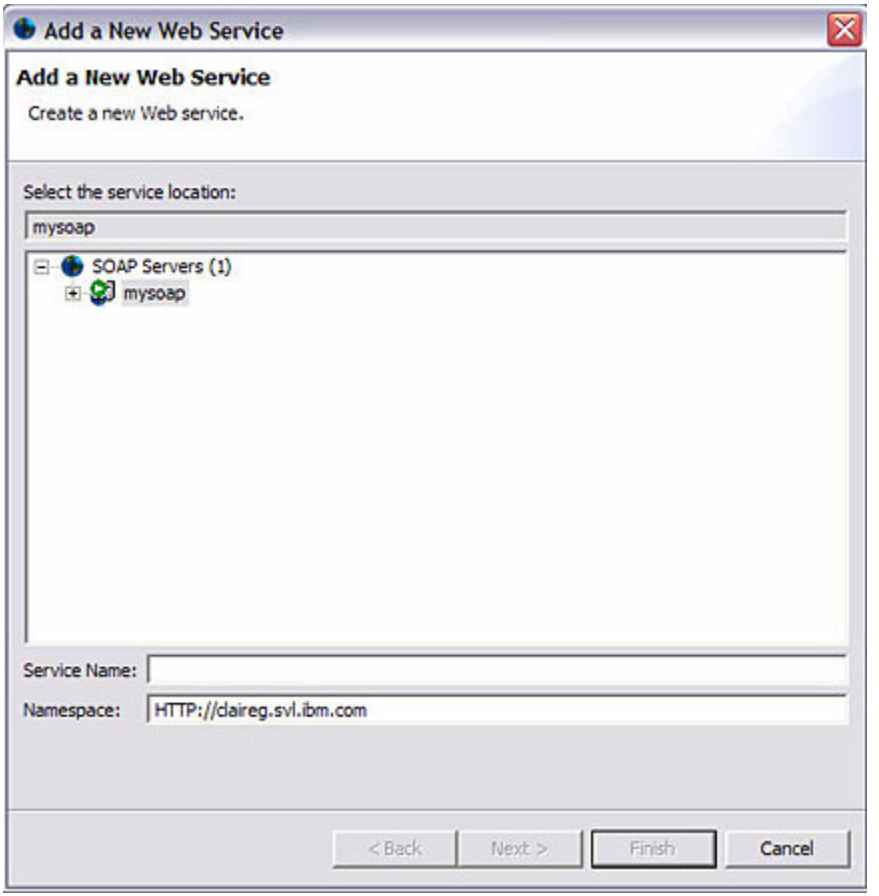


Creating a Subroutine Web Service Using the Wizard

To create a Query Web Service using the wizard, right-click the SOAP server where you want to creat the service, then click **Create New Web Service**, as shown in the following example:



The **Add a New Web Service** dialog box appears, as shown in the following example:

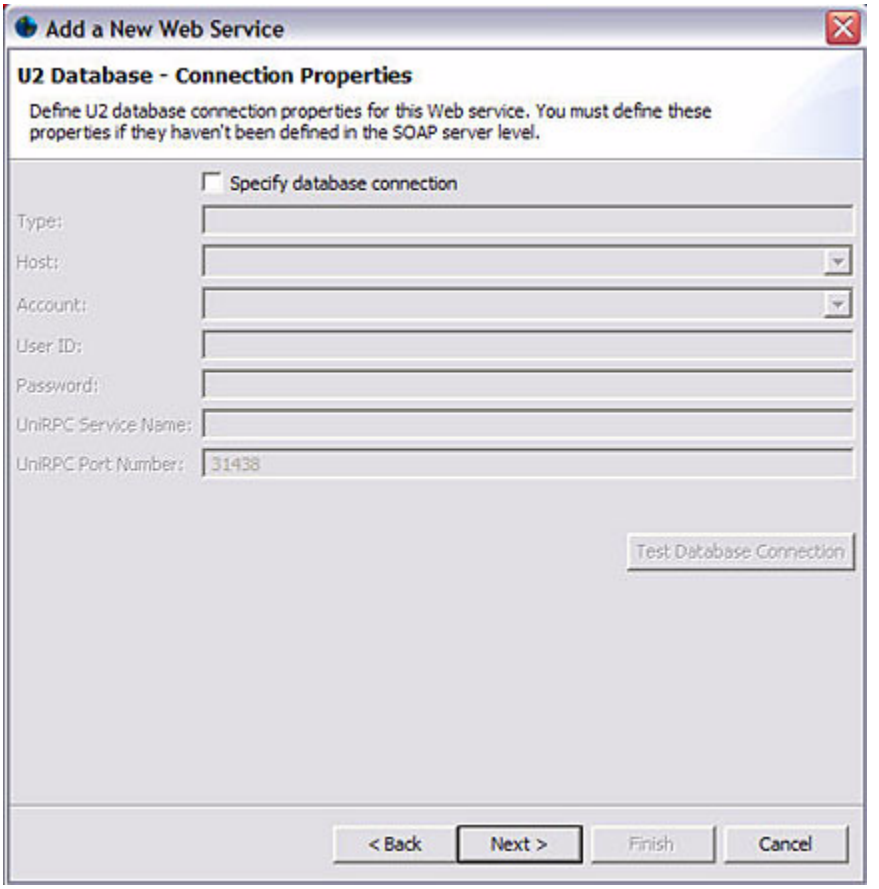


Enter the name of the service in the **Service Name** box.

Verify or enter the namespace for the service in the **Namespace** box.

Click **Next**.

The **U2 Database - Connection Properties** dialog box appears, as shown in the following example:



If you have not defined the database connection properties at the SOAP Server-level, you must define them for this Web Service.

Select the **Specify database connection** check box.

Type

The type of database connection. At this release, the only type of connection supported is “fixed.”

Host

Select the name of the host server from the list of UniVerse servers you have defined. The server should be running.

Account

Select the account name on the UniVerse server you specified where you want attach when you connect from the list. This account must contain the data files you are accessing with the subroutine.

User ID and Password

In the **User ID** box, enter your log on name for the server. Enter the corresponding password in the **Password** box.

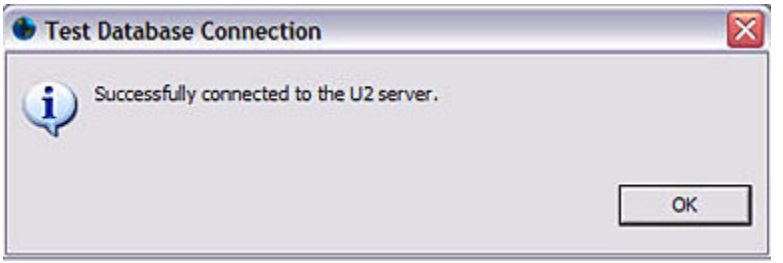
UniRPC Service Name

Enter the appropriate UniRPC Service Name in the**UniRPC Service Name** box. For UniVerse, the service name is uvcs. For UniData, the service name is udcs.

UniRPC Port Number

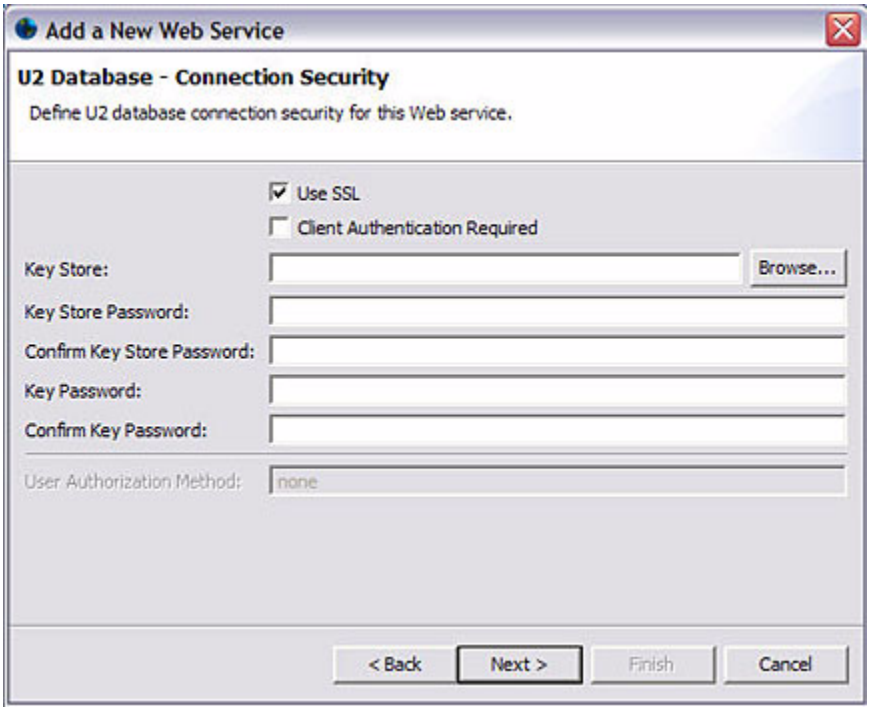
Enter the port number of the UniRPC server running on the host in the **UniRPC Port Number** bos. The default port number is 31438.

To test the connection to the database, click **Test Database Connection**. If the connection is successful, the following message appears:



Click **Next.**

The **U2 Database - Connection Security** dialog box appears, as shown in the following example:



Key Store

Enter the full path to the key store on the SOAP server, or click **Browse** to navigate to the key store.

Key Store Password

Enter the password corresponding to the key store you defined in the **Key Store** box.

In the **Confirm Key Store Password** box, reenter the password.

Key Password

Enter the encryption key password, if one exists, in the **Key Password** box. Reenter the password in the **Confirm Key Password** box.

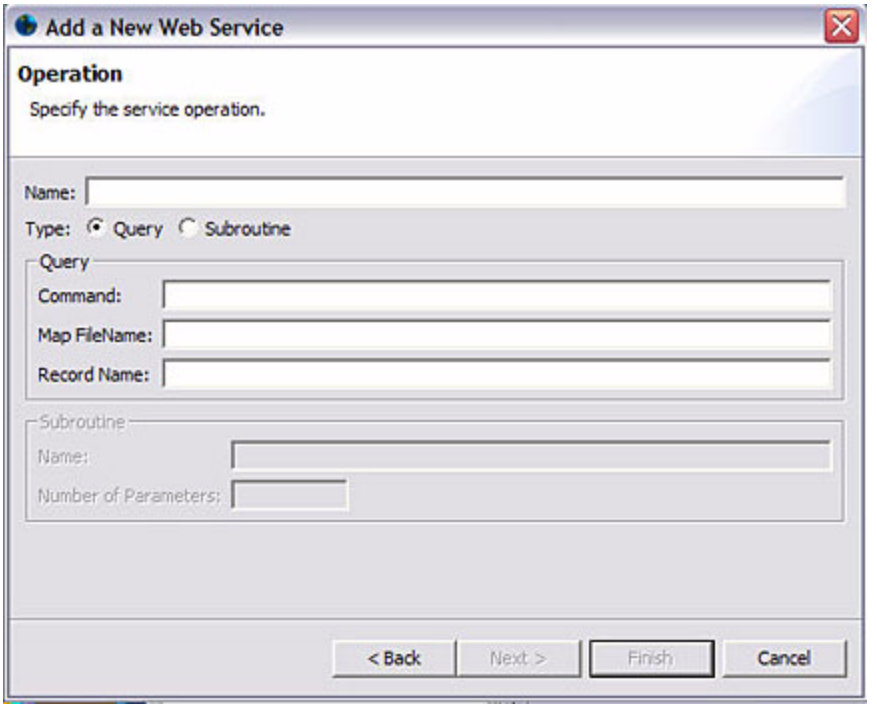
Enable Authentication

If you want the client to send its certification for authentication, select the **Client Authentication Required** check box.

For detailed information about SSL, see *UniVerse Security Features*.

Click **Next**.

The **Operation** dialog box appears, as shown in the following example:t

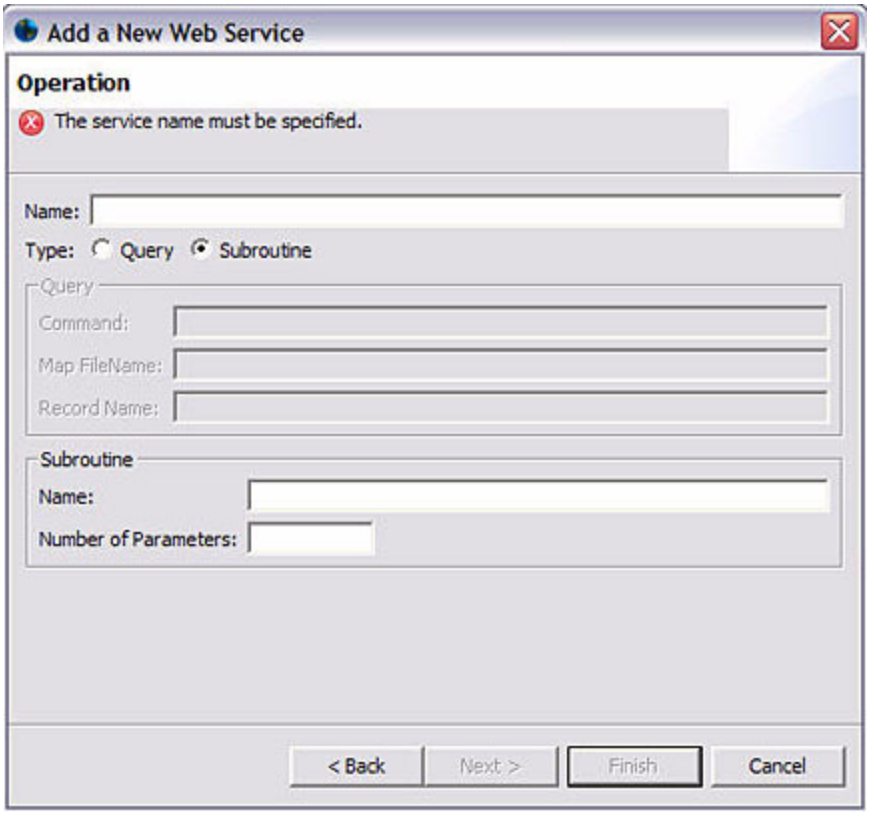


Name

Enter the name for the Web Service you are creating in the **Name** box.

Type

Select **Subroutine** as the type of Web Service you are creating. The **Operation** dialog box appears, as shown in the following example:



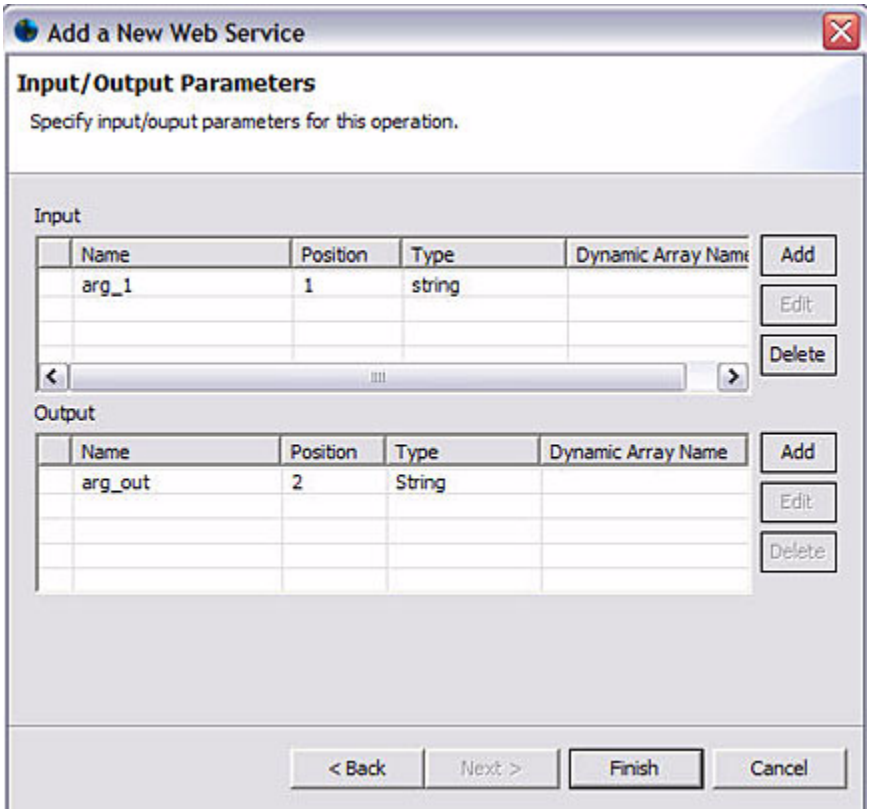
Define Subroutine

Enter the name of the subroutine in the **Subroutine Name** box. Enter the total number of parameters defined for the subroutine in the **Number of Parameters** box. In the example subroutine, we have two parameters, one input, and one output.

Click **Next**.

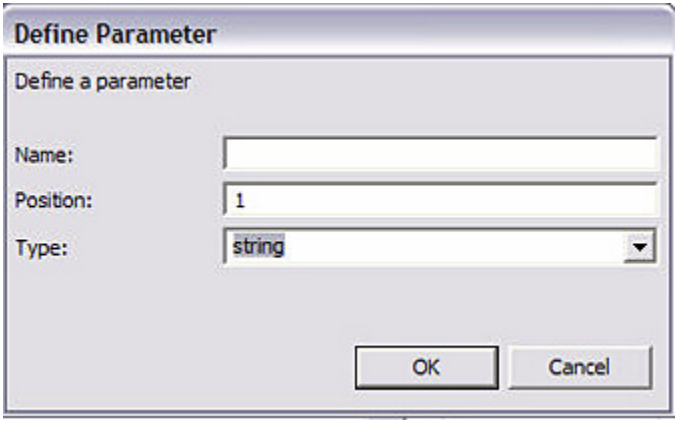
Define Input/Output Parameters

The **Input/Output Parameters** dialog box appears, as shown in the following example:



Define Input Parameters

To define the input parameters for the subroutine, highlight the parameter in **Input** area of the **Subroutine Details** dialog box, then click **Edit**. The **Define Parameter** dialog box appears:



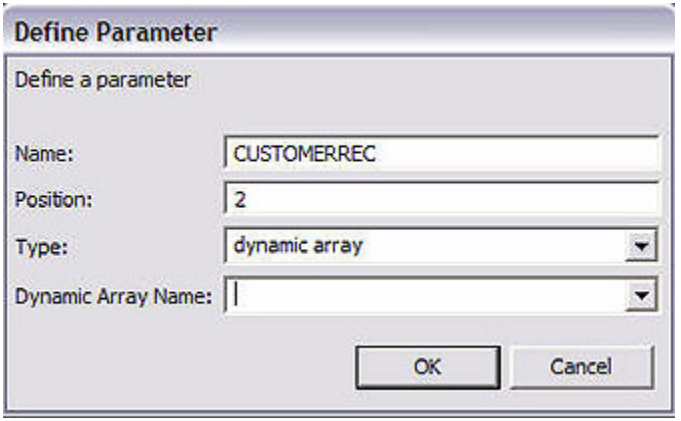
In the example subroutine, the input parameter is the Customer ID, and it is the first parameter in the subroutine.

In the **Name** box, enter a meaningful name for the parameter. This name does not have to be the same as the one defined in the UniVerse BASIC program. Enter the position the input parameter appears in the subroutine in the **Position** box. In the **Type** box, enter the data type for the input parameter, the click **OK**.

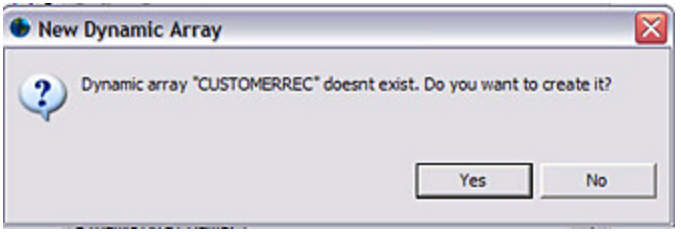
Define Output Parameters

To define the output parameters for the subroutine, highlight the parameter the IBM U2 Web Services Developer populated in **Output** area of the **Subroutine Details** dialog box, then click **Edit**. The **Define Parameter** dialog box appears.

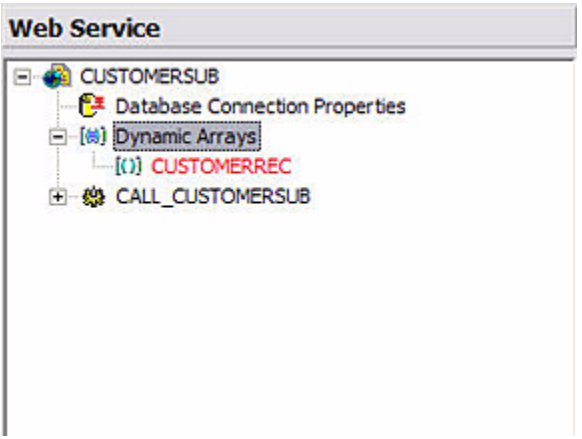
Enter the name of the output parameter in the **Name** box. Enter the position of the output parameter in the subroutine in the **Position** box. Enter the data type of the output parameter in the **Type** box. In our example, the output parameter is a dynamic array.



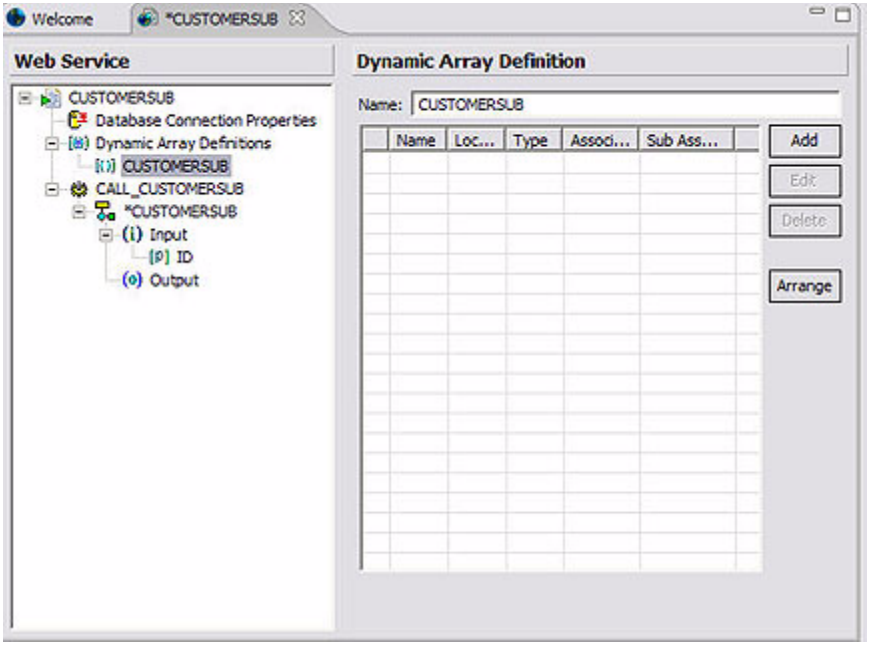
Enter the name of the dynamic array in the **Dynamic Array Name** box. You can choose an existing dynamic array, or create a new one. If the dynamic array you specify does not exist, a message similar to the following example appears:



Click **Yes** to define a new dynamic array. The name of the array appears under the **Dynamic Arrays** area of the **Web Services** window, as shown in the following example:

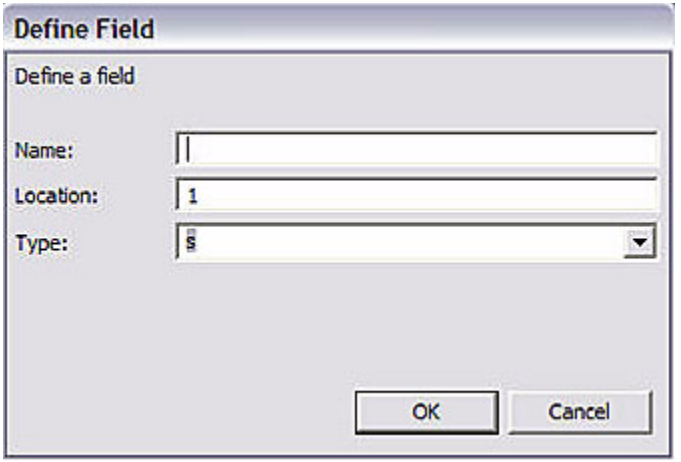


Double-click the dynamic array for which you want to define fields. The **Dynamic Array Fields** dialog box appears, as shown in the following example:



You can enter the dynamic arrays manually, or populate them using a drag-and-drop operation.

To manually add a field, click **Add**. The **Define Field** dialog box appears, as shown in the following example:



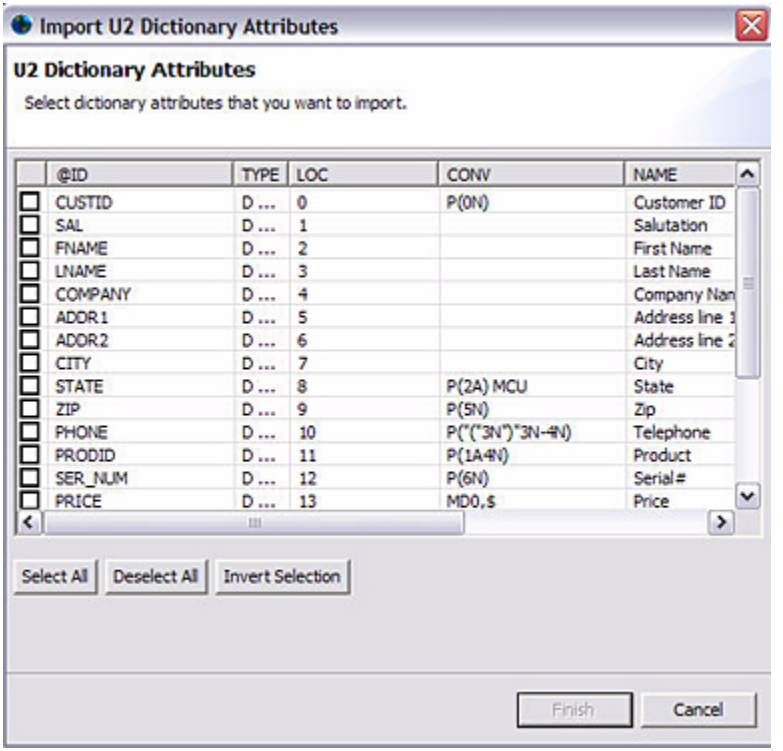
Enter the name of the field in the dynamic array in the **Name** box.

Enter the location of the field in the dynamic array in the **Location** box.

Enter the type of field in the **Type** box. Valid types are:

- s – Singlevalued
- mv – Multivalued
- ms – Multi-subvalued

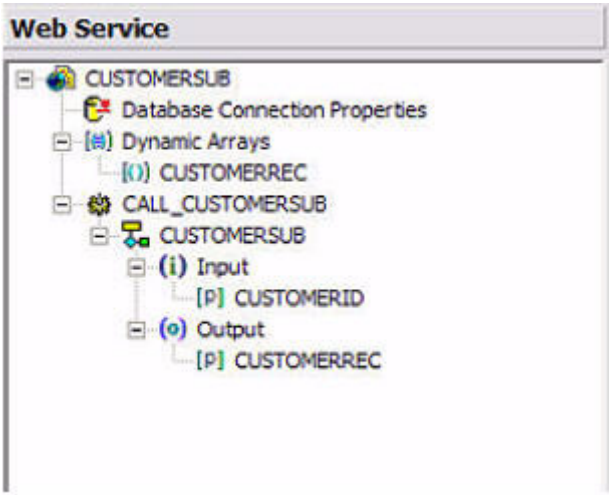
To populate the fields, from the Database Files in the account for which you are creating the web service, move the file pertaining to the dynamic array using a drag-and-drop operation. The IBM U2 Web Services Developer populates the **U2 Dictionary** dialog box with each D-type dictionary record, as shown in the following example:



Select the check boxes next to the dictionary record ID you want to include in the dynamic array. The number of attributes you select must match the number of fields in the parameter in the subroutine. In our example, we are returning the entire record, so each dictionary attribute is selected, except for the CUSTID.

Enter CTRL+S or the **Save** icon to save your dynamic array definition.

When you have finished defining the parameters of the subroutine, the input parameters, output parameters, and dynamic arrays appear in the **Web Services** area of the **IBM U2 Web Services Developer** window, as shown in the following example:



Executing the Web Service

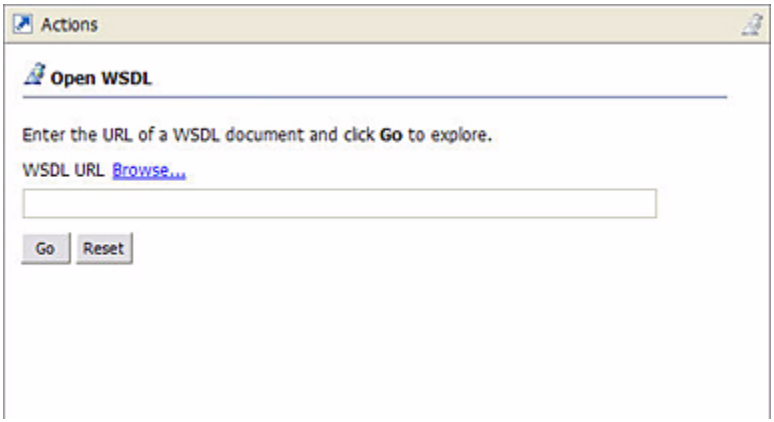
To execute the Web Service, click the **Launch** icon from the toolbar:

Launch icon

The **Web Services Explorer** window appears. Click the **WSDL** icon, as shown in the following example:

WSDL icon

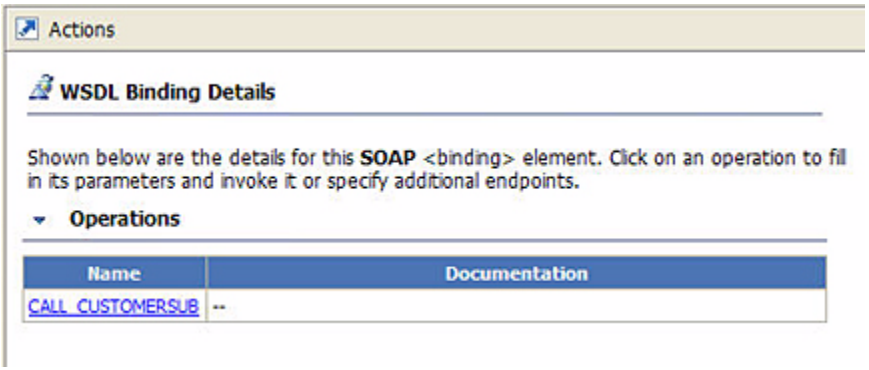
Under the **Navigator** area of the **Web Services Explorer** window, click **WSDL Main**. The **Open WSDL** dialog box appears, as shown in the following example:



From the **U2 Web Service** area of the **IBM U2 Web Services Developer** window, click the web service you want to execute. The properites of the web service appear in the **Properties** area of the **Web Service** window, as shown in the following example:

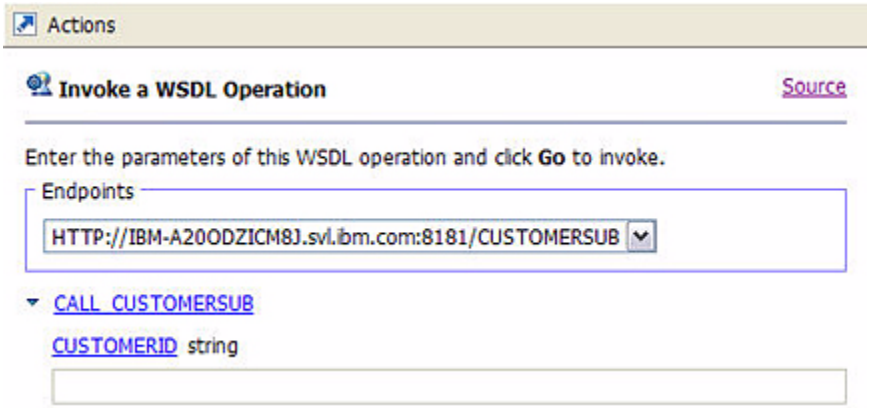
In the WSDL URL dialog box, enter the value of the **End Point** property, then click **Go**.

In the **WSDL Details** window, click the operation you want to execute, as shown in the following example:

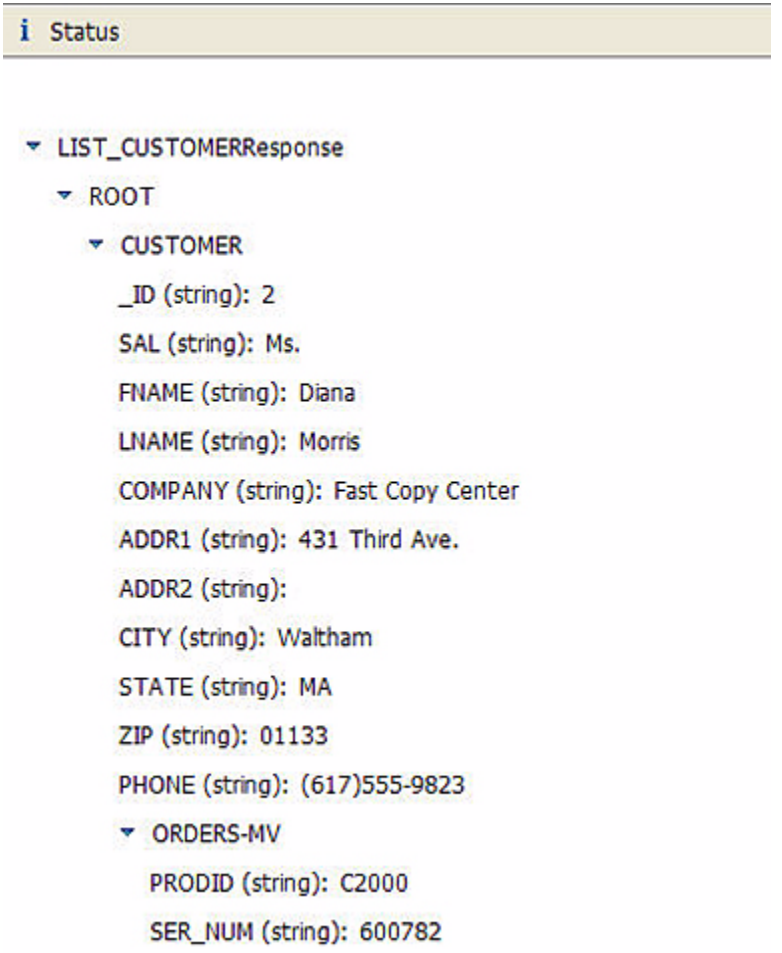


Click the operation you want to execute, in this case, CALL CUSTOMERSUB.

The Web Service operation prompts you for input parameters you previously defined, as shown in the following example:



Enter the input value, then click **GO**. The IBM U2 Web Services Developer displays the resulting XML document in the **Status** area of the **Invoke a WSDL Operation** dialog box, as shown in the following example:



Chapter

5

Miscellaneous Features

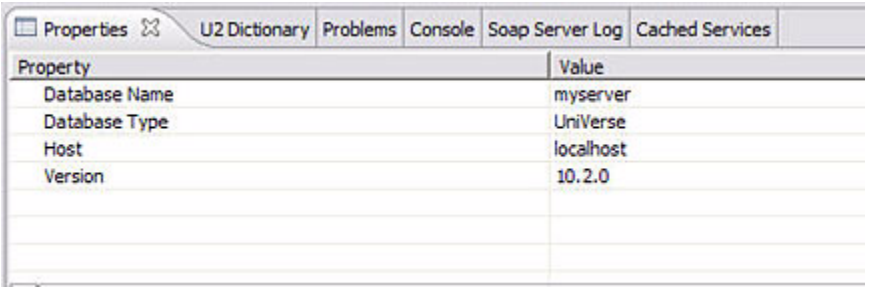
This chapter describes miscellenous features of the IBM U2 Web Services Developer.

Displaying Properties

You can display the properties of a server, account or file in the **Properties** area to of the IBM U2 Web Services Developer window.

Displaying Server Properties

To display the properties of a server, click the server for which you want to display properties. The properties of that server appear in the **Properties** are of the IBM U2 Web Services Developer window, as shown in the following example:



The **Database Name** is the name of the server you created.

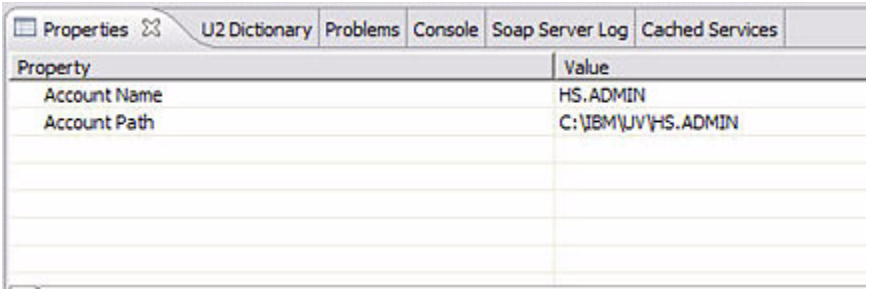
The **Database Type** is the type of database to which you are connected. Valid types are UniVerse or UniData.

The **Host** is the host name.

The **Version** is the version number of the **Database Type** to which you are connected.

Displaying Account Properties

To display the properties of an account, click the account for which you want to display properties. The properties of that account appear in the **Properties** area of the IBM U2 Web Services Developer window, as shown in the following example:

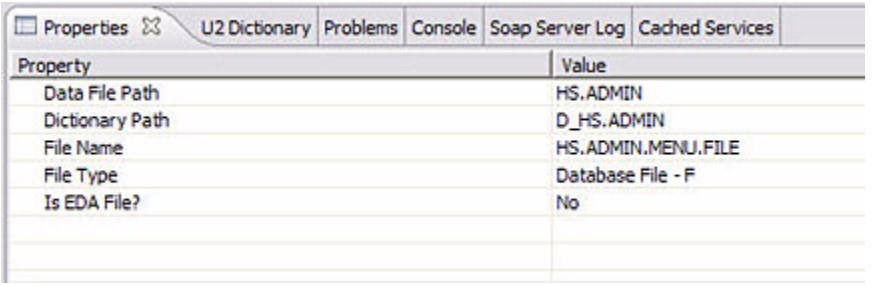


The **Account Name** is the name of the account on the server.

The **Account Path** is the full path to the account on the server.

Displaying File Properties

To display the properties of a file, click the account where the file resides, then click the file for which you want to display properties The properties of that file appear in the **Properties** area of the IBM Web Services Developer window, as shown in the following example:



The **Data File Path** is the path to the file. If the file resides locally, this field displays the name of the file. If the file resides in a remote location, this field displays the full path to the file.

The **Dictionary Path** is the name of the dictionary for the file you selected.

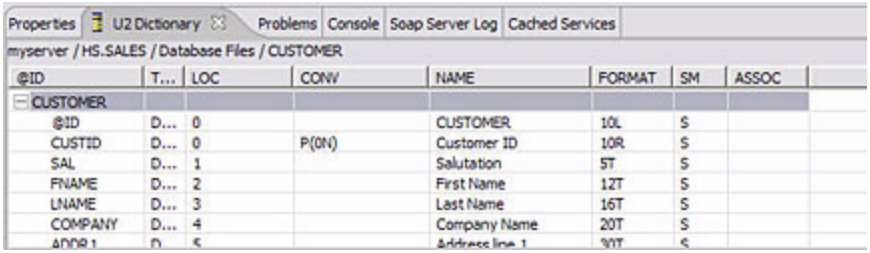
The **File Name** is the name of the file.

The **File Type** is the type of file.

Is EDA File indicates whether the file is an EDA file or not. This option is only valid on a UniData database.

Displaying File Dictionaries

To display the dictionary or a file, click the file for which you want to display the dictionary, the click the **U2 Dictionary** tab. Information similar to the following example appears:

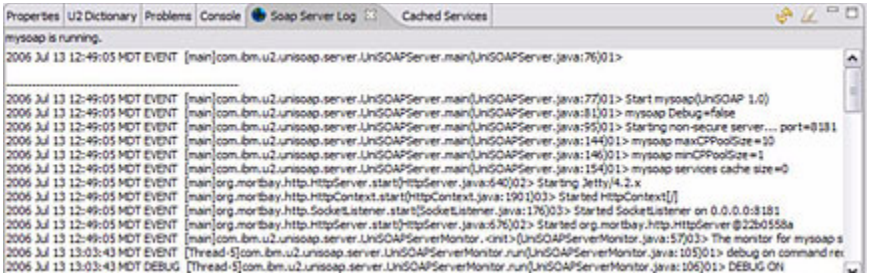


IBM U2 Web Services Developer displays each record in dictionary, with the following information:

- Dictionary **Record ID.**
- Type of dictionary record.
- Field location of the dictionary record if a D-type record, formula if an I-type record, or association components if a PH-type record.
- Conversion formula, if specified.
- Display name, if specified.
- Format code.
- Singlevalue or multivalued code.
- Association name, if part of an association.

Displaying SOAP Server Logs

Click **Soap Server Log** to display the SOAP Server Logs, as shown in the following example:



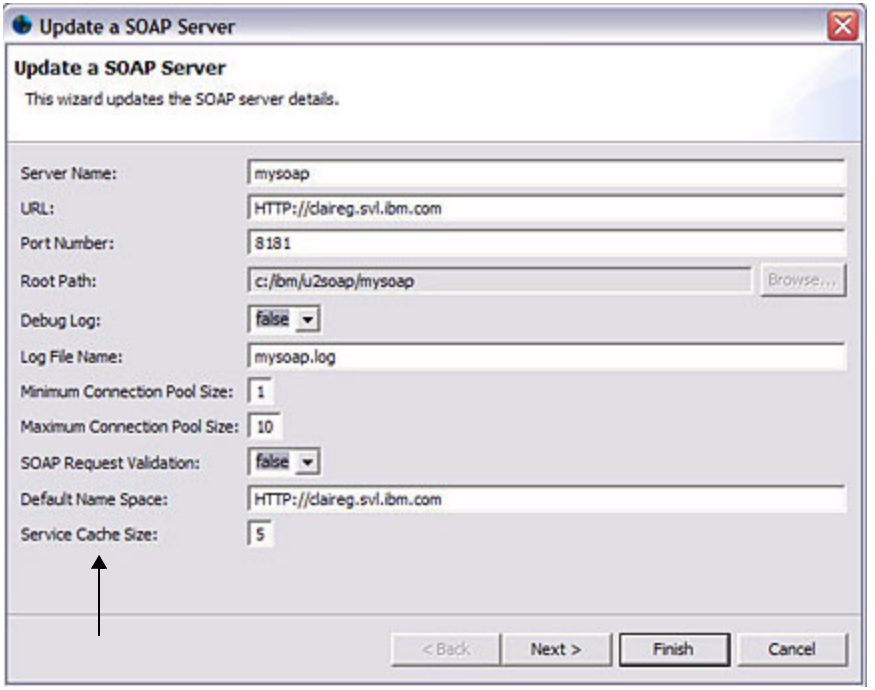
IBM U2 Web Services developer displays the last 64KB in the SOAP Server log.

To erase the contents of the log file click the eraser icon, as shown in the following example:

To refresh the contents of the log file, click the circular arrows icon, as shown in the following example:

Displaying Cached Services

When you define the properties for your SOAP Server, you can specify the Server Cache Size, as shown in the following example:



For performance purposes, you can set this value to number of web services currently running, or larger. If you do not set this value, the SOAP Server reloads the web service each time. If you do set this value, next time a web service is needed, the SOAP Server reads it from cache.

If you are developing a web service, we recommend keeping this value at 0. Otherwise, the SOAP Server reloads the cache each time the web service is changed.

Select the **Cached Services** tab to display the web services currently loaded in cache.

Chapter

6

Accessing the Web Services Programatically

The chapter describes how to access Web Services programatically.

A defined Web Service is identified by its URL. The format of the URL is:

SOAPserverURL/virtual_directory_path/servicename

SOAPserverURL is defined when you create a new SOAP server.

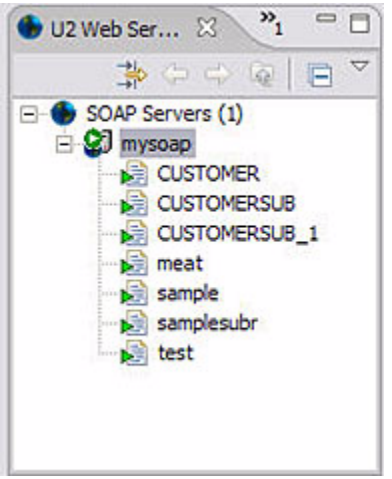
virtual_directory_path is a path relative to the configurable root directory of the SOAP server.

servicename is the name of the web service.

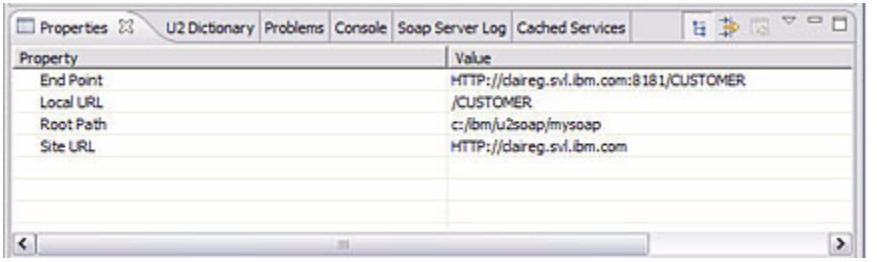
You can identify the URL for the Web Service from the IBM U2 Web Services Developer.

Viewing the Web Service URL

From the IBM U2 Web Services Developer main window, expand the Web Server where the web service is located, as shown in the following example:



Click the web service for which you want to view the URL. Click the **Properties** tab. The URL for the web service appears in the **End Point** field, as shown in the following example:



View the WSDL File

To view the WSDL file for a web service, enter the URL in a browser. The contents of the WSDL file appears, as shown in the following example:

Programming a Web Services Client

To write a Web Services client program, you create an object of a proxy class, then call the methods for that object.

Many software tools can accept a WSDL file and generate proxy classes in many kinds of languages, including Java, C#, VB.NET, and so forth.

Proxy classes can be generated automatically because the definition contained in a WSDL file gives a complete description of the interfaces to the Web service you define, including the XML schemas of the input and output data, the SOAP bindings to these schemas, the transportation protocol, and so forth.

The proxy classes include a service proxy class, as well as several classes that represent data structures used in the Web Services. These classes are self-explanatory.

Generating a Client Proxy in IBM Websphere Application Developer

To generate a Web Service client proxy in IBM Websphere Application Developer, select **New** from the menu, choose Web service client and the project type, then follow the instructions from the wizard. Provide the URL of the WSDL file. For more information, see the documentation for the IBM Websphere Application Developer.

Generating a Client Proxy in Visual Studio.Net

To generate a Web Service client proxy in Visual Studio.NET, start a new project, then add a Web Reference. The Add Web Reference wizard prompts for the URL for the WSDL file. After providing the URL, the wizard generates a set of proxy classes for the current project. For more information, see the documentation or Visual Studio .NET.

Chapter

7

Deploying Web Services

The U2 Web Services Developer deployment feature enables you to select a SOAP server defined locally and generate a deployment package, in the form of a zip file, that contains the following:

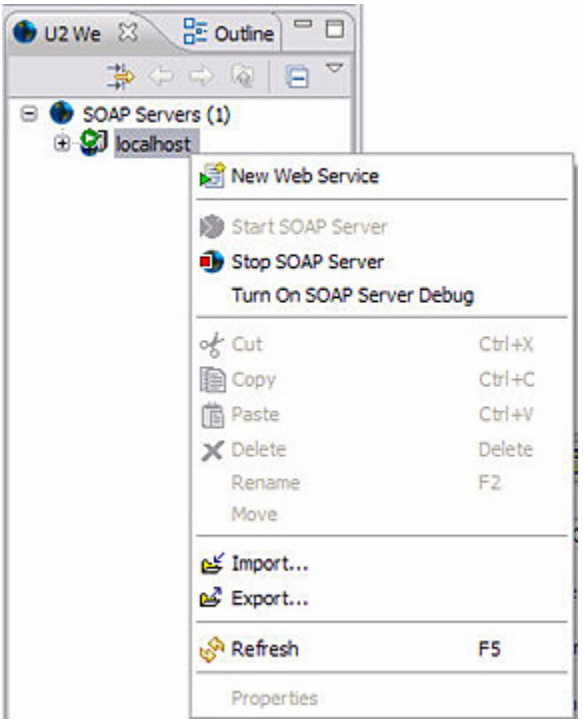
- WSDL files
- The runsoapserver and stopsoapserver scripts
- All required files for a runtime SOAP server

The zip file contains both a .bat and .sh script which enables you to deploy your web services on any platform.

You must be running JRE 1.5 or above to use the U2 Web Services Development deployment feature.

Exporting Web Services

To create the zip file to deploy your web services, right-click the SOAP server which contains the web service you want to deploy, as shown in the following example:

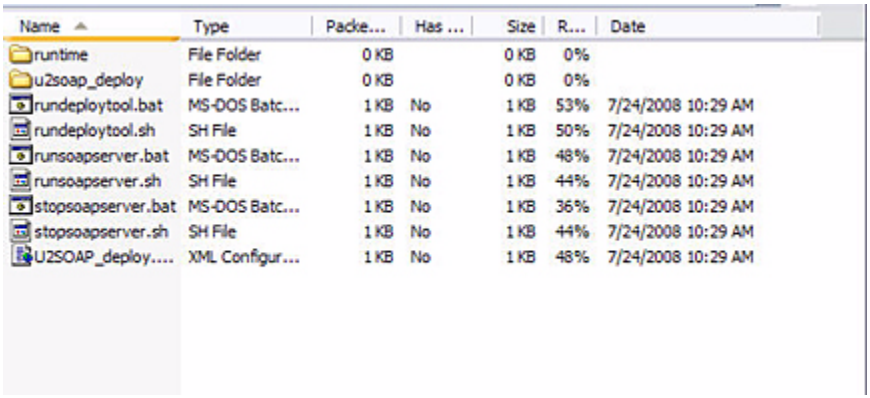


Click **Export**. The **Export SOAP Servers** dialog box appears, as shown in the following example:

In the **Available SOAP Servers** section, select the SOAP servers you want to export.

Enter the path for the zip file you want to create in the **Archive File** box, or click **Browse** to search for the location. By default, the U2 Web Services developer places the zip file in C:\IBM\U2Tools*soapservername*.zip.

The following example shows sample contents of the zip file after the U2 Web Services Developer exports the SOAP server:



Name	Type	Packe...	Has ...	Size	R...	Date
runtime	File Folder	0 KB		0 KB	0%	
u2soap_deploy	File Folder	0 KB		0 KB	0%	
rundeploytool.bat	MS-DOS Batc...	1 KB	No	1 KB	53%	7/24/2008 10:29 AM
rundeploytool.sh	SH File	1 KB	No	1 KB	50%	7/24/2008 10:29 AM
runsoapserver.bat	MS-DOS Batc...	1 KB	No	1 KB	48%	7/24/2008 10:29 AM
runsoapserver.sh	SH File	1 KB	No	1 KB	44%	7/24/2008 10:29 AM
stopsoapserver.bat	MS-DOS Batc...	1 KB	No	1 KB	36%	7/24/2008 10:29 AM
stopsoapserver.sh	SH File	1 KB	No	1 KB	44%	7/24/2008 10:29 AM
U2SOAP_deploy....	XML Configur...	1 KB	No	1 KB	48%	7/24/2008 10:29 AM

Deploying the SOAP Server

Unzip the zip file on the target machine. Review the rundeploytool.bat or rundeploytool.sh file to verify that the path to javaw is correct.

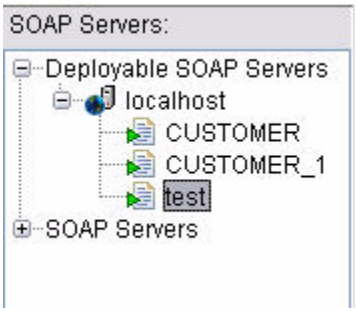
From the directory where you unzipped the file, execute the rundeploytool command.

rundeploytool *source_directory target_directory*

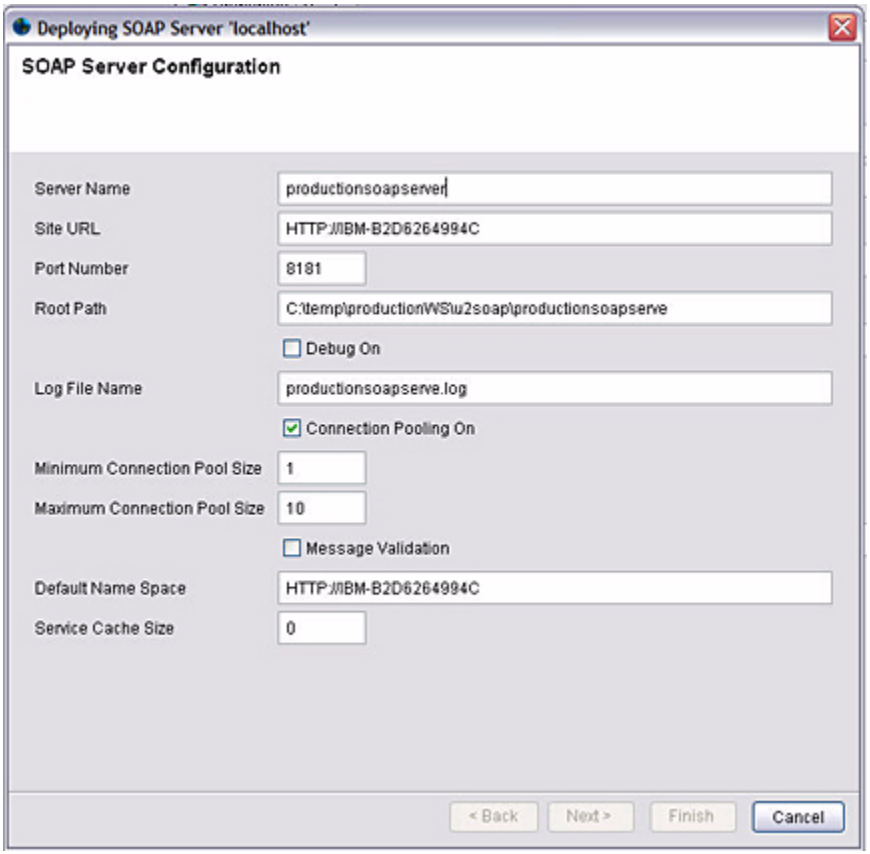
For example:

rundeploytool . C:\temp\myservice

Expand **Deployable SOAP Servers**, right-click the SOAP server you want to deploy, then click **Deploy**, as shown in the following example. If the SOAP server already exists, click **Deploy Into**.



The **U2 SOAP Service Deployment** dialog box appears, as shown in the following example:



You can specify the environment for the new SOAP server in the U2 SOAP Service Deployment dialog box, including:

- The SOAP server configuration
- The SOAP server connection security
- The database connection properties
- The database connection security

The Deployment tool populates the window with information from your local machine. You can change this information if necessary.

Server Name

In the **Server Name** box, enter a unique name for the SOAP server.

Site URL

The URL for the SOAP server you specify. Make sure the URL you specify is valid and accessible.

Port Number

The port number on which the server will listen.

Root Path

The path to the root directory where the definitions to the web service are stored.

Debug On

Select the **Debug On** check box if you want to capture debugging information. If you select this check box, the U2 IBM Web Services Developer starts the debug log each time you start the server.

Log File Name

The name of the debug log.

Connection Pooling On

The **Connection Pooling On** setting determines if the U2 Soap Server uses connection pooling to connect to UniVerse. The term *connection pooling* refers to the technology that pools permanent connections to data sources for multiple threads to share. It improves application performance by saving the overhead of making a fresh connection each time one is required. Instead of physically terminating a connection when it is no longer needed, connections are returned to the pool and an available connection is given to the next thread with the same credentials.

Connection Pooling in enabled by default. If you do not want to use Connection Pooling, clear the **Connection Pooling On** check box.

***Note**: IBM recommends using Connection Pooling for superior performance and scability.*

Connection Pool Size

You can set the minimum and maximum size of the connection pool. If you do not define these sizes, the minimum size defaults to 1 and the maximum size defaults to 10. The minimum size determines the initial size of the connection pool.

The size of the connection pool changes dynamically between the minimum and maximum sizes you specify, depending on the system demands. When there are no pooled connections available, the U2 Soap Server either creates another connection, if the maximum connection pool size has not been reached, or keeps the thread waiting in the queue until a pooled connection is released or the request times out. If a pooled connection is idle for a specified time, it is disconnected.

Message Validation

Checks the SOAP request again the schema defined in the WSDL file. This option is time consuming. By default, message validation is not selected.

Default Name Space

The name space for the Web Services you define.

Server Cache Size

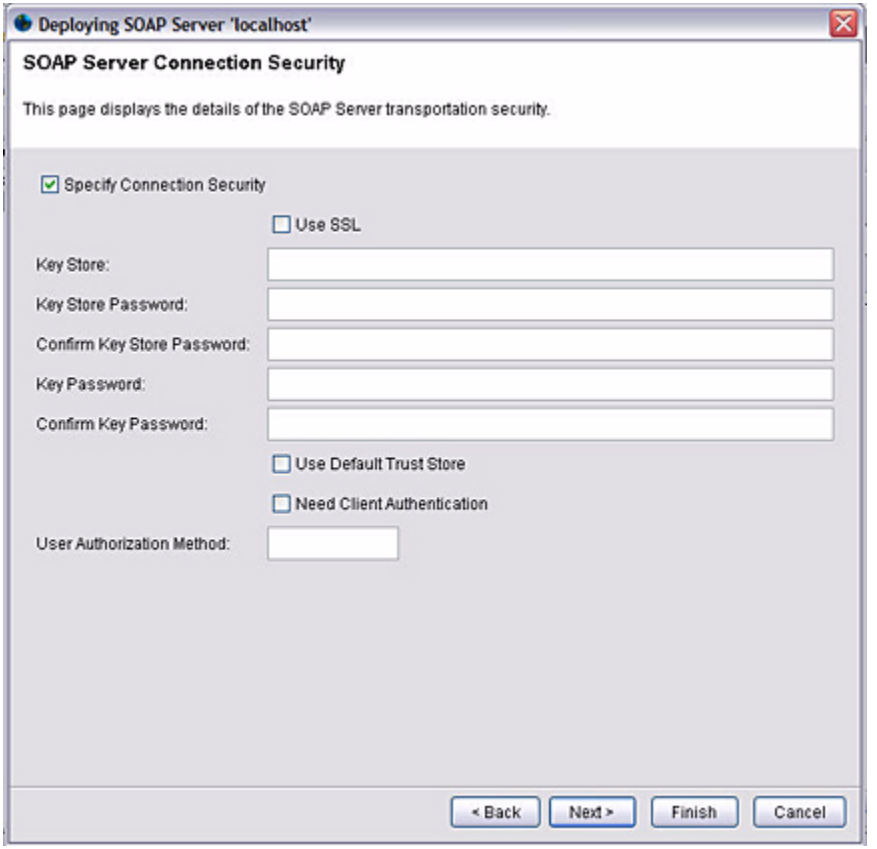
For performance purposes, you can set this value to a number greater than 0 to indicate the number of web service definitions you want to keep in the cache. If you set this value, the SOAP Server will always try to read the web service definition from the cache first. If you do not set this value, the SOAP Server reloads the web service each time from disk.

If you are developing a web service, we recommend keeping this value at 0. This setting forces the SOAP Server to reload the new web service definition each time. Select the **Cached Services** tab to display the web services currently loaded in cache.

Click **Next**.

Define Security between the Client and the SOAP Server

Click the **Specity Connection Security** check box if you want to define connection security between the client and the SOAP server. The **SOAP Server Connection Security** dialog box appears, as shown in the following example:



Use SSL

Select the **Use SSL** check box if you want to define SSL security parameters.

Key Store

Enter the full path to the key store on the SOAP server.

Key Store Password

Enter the password corresponding to the key store you defined in the **Key Store** box.

In the **Confirm Key Store Password** box, reenter the password.

Key Password

Enter the encryption key password, if one exists, in the **Key Password** box. Reenter the password in the **Confirm Key Password** box.

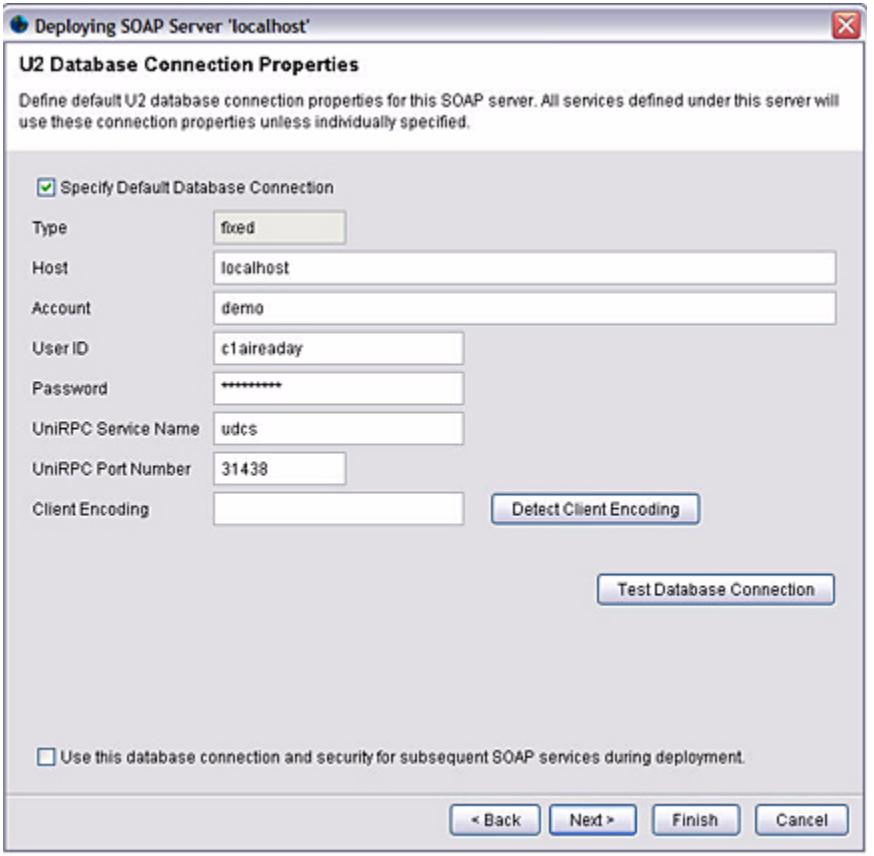
Enable Authentication

If you want the client to send its certificate for authentication, select the **Need Client Authentication** check box.

Click **Finish**.

Set Connection Properties

Click the **DB Connection** tab to define the connection properties for the SOAP server. The **U2 Database - Connection Properties** dialog box appears, as shown in the following example:



The deployment tool populates the fields in the **U2 Database Connection Properties** dialog box based on the information contained in the deployable SOAP server. You can change this information if necessary.

Type

The type of database connection. At this release, the only type of connection supported is “fixed.”

Host

Enter the name of the host server. The server should be running.

Account

Enter the account name on the server you specified where you want attach when you connect to the server. This account must contain the data files you are accessing with the web service.

User ID and Password

In the **User ID** box, enter your log on name for the server. Enter the corresponding password in the **Password** box.

UniRPC Service Name

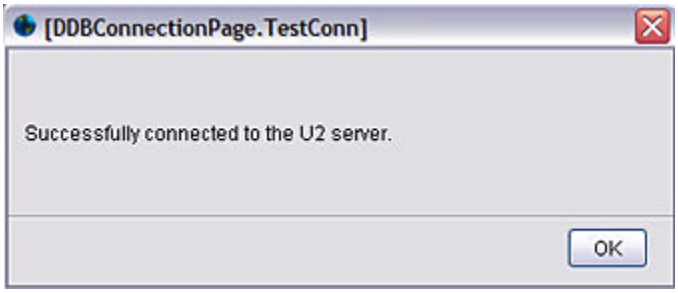
Enter the appropriate UniRPC Service Name in the**UniRPC Service Name** box. For UniVerse, the service name is uvcs.

UniRPC Port Number

Enter the port number of the UniRPC server running on the host in the **UniRPC Port Number** box.

Test Database Connection

To ensure that you can connect to the database, click **Test Database Connection**. If the connection is successful, the following message appears:



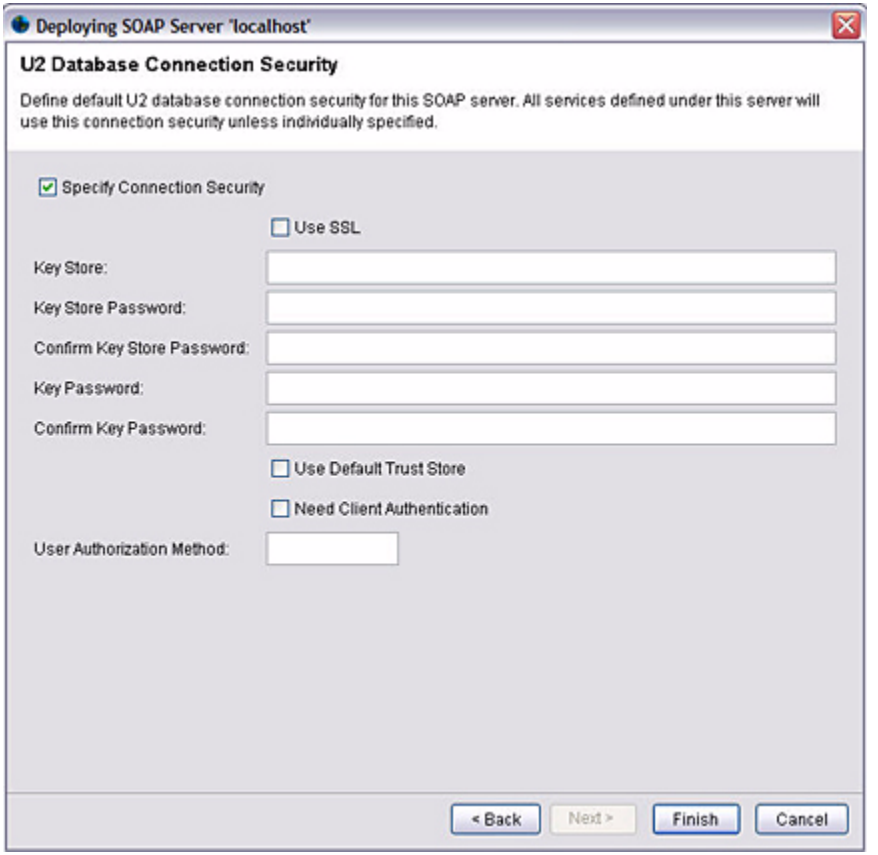
Use Connection Properties for Subsequent Servers

If you want to use the connection properties you specified for subsequent SOAP servers you deploy, select the **Use this database connection and security for subsequent SOAP services during deployment** check box.

Click **Next.**

Define UniVerse Database Connection Security

Click the **DB Security** tab if you want to define security between the SOAP server and the UniVerse database. The **U2 Database Connection Security** dialog box appears, as shown in the following example:



Key Store

Enter the full path to the key store on the SOAP server, or click **Browse** to navigate to the key store.

Key Store Password

Enter the password corresponding to the key store you defined in the **Key Store** box.

In the **Confirm Key Store Password** box, reenter the password.

Key Password

Enter the encryption key password, if one exists, in the **Key Password** box. Reenter the password in the **Confirm Key Password** box.

Use Default Trust Store

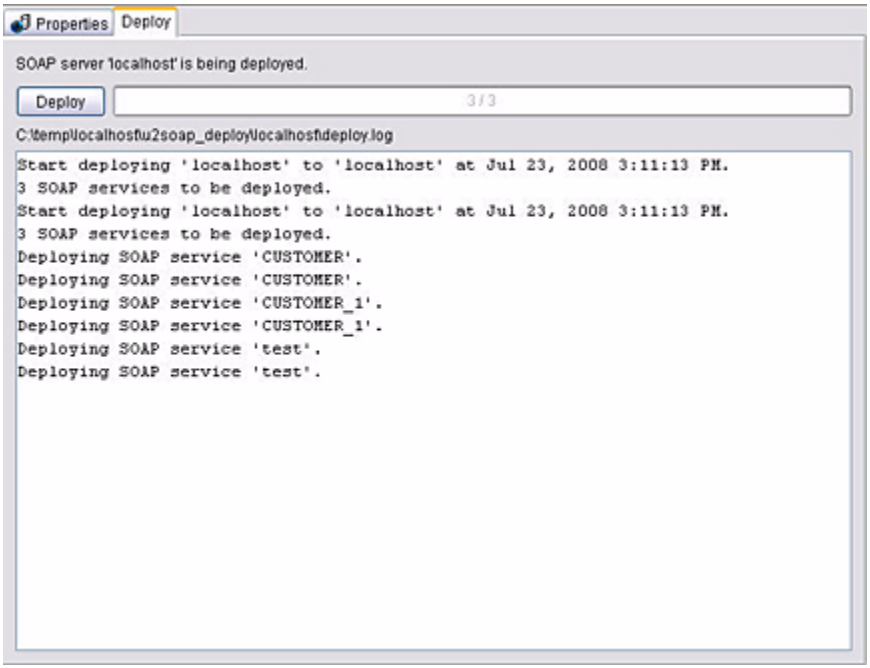
Enable Authentication

If you want the client to send its certificate for authentication, select the **Need Client Authentication** check box.

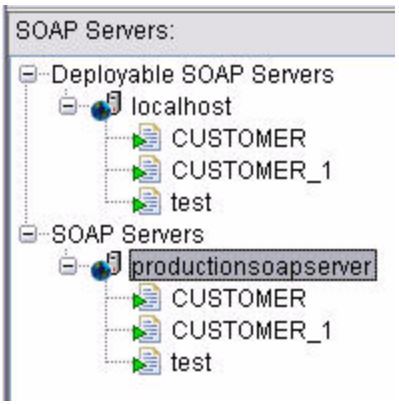
Click **Next**.

For detailed information about SSL, see *UniVerse Security Features*.

U2 Web Services Developer deploys the web services you specified. When deployment is complete, a window similar to the following example appears:



The deployable web services appear in the SOAP Servers area of the **U2 SOAP Service Deployment** dialog box, and shown in the following example:



Running and Stopping the SOAP Server

To run the SOAP server, in the target directory enter:

runsoapserver *soap_server_name*

To stop the SOAP server, in the target directory enter:

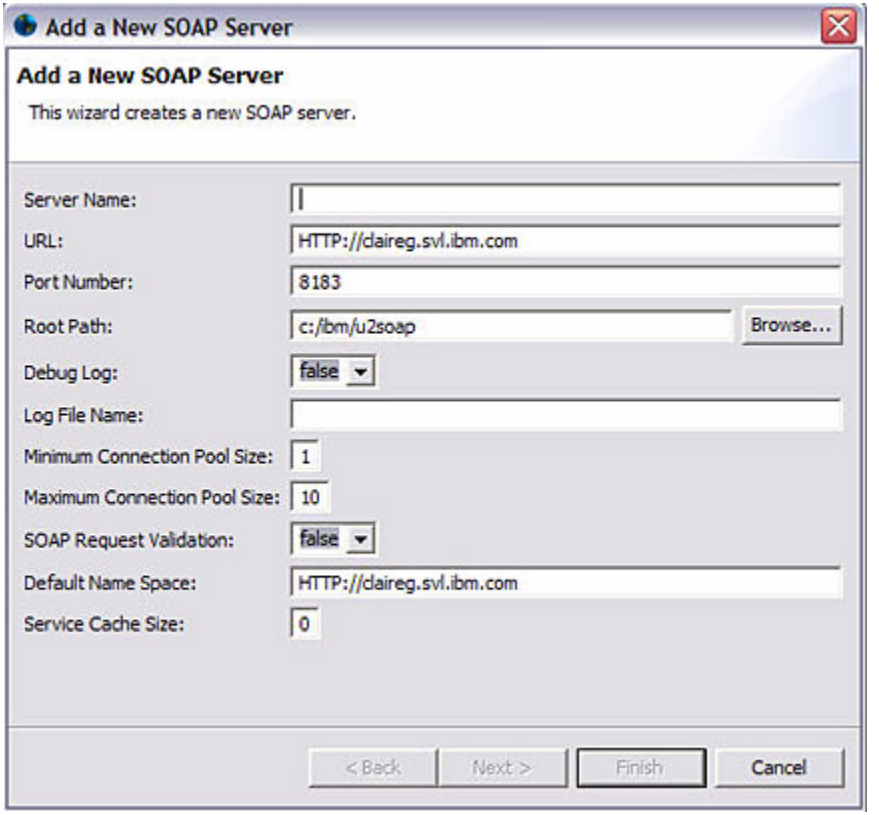
stopsoapserver *soap_server_name*

Monitoring a Remote SOAP Server

Through the U2 Web Services Developer, you can monitor a remote SOAP server. Complete the following steps to set up remote monitoring:

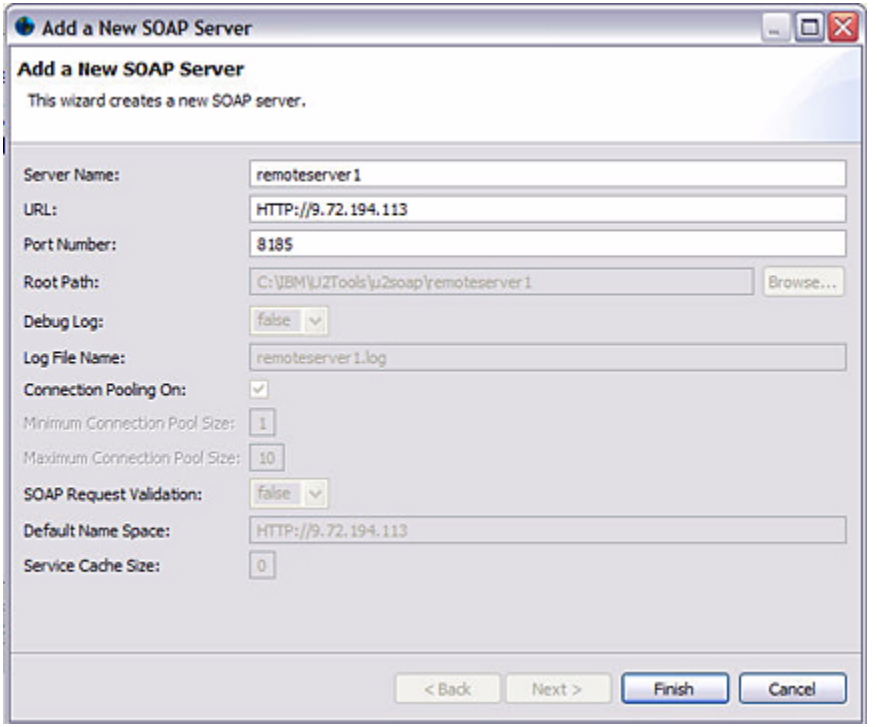
Define the Remote Server

Right-click SOAP Servers, then click **New SOAP Server**. The **Add a New SOAP Server** dialog box appears, as shown in the following example:



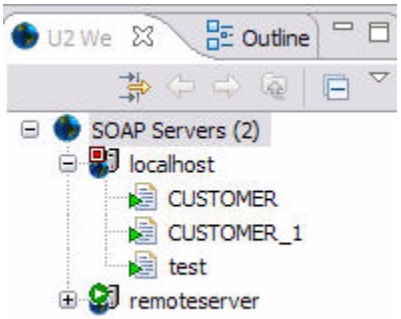
Enter a name for the remote server in the Server Name box.

Enter the URL for the remote server in the URL box. Based on this URL, U2 Web Services Developer detects that the server is remote, and dims unavailable options, as shown in the following example:



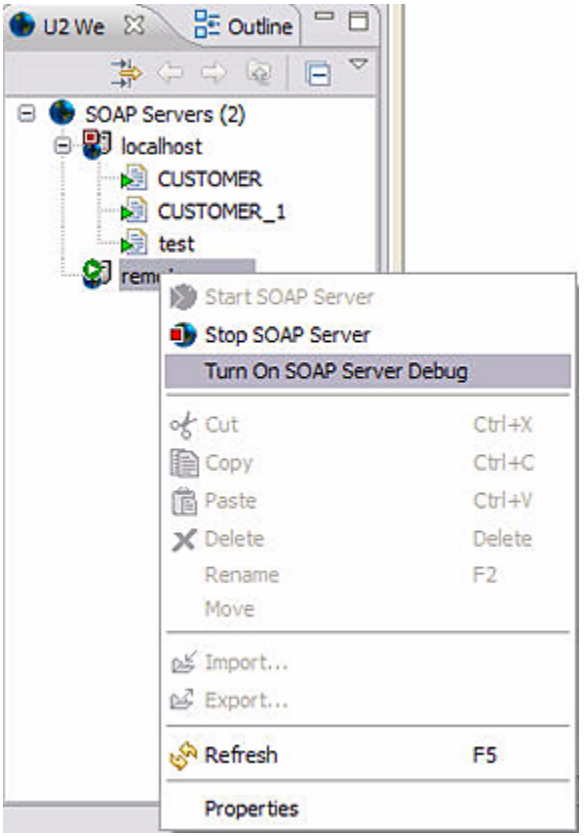
Enter the port number for the remote server in the **Port Number** box.

If the remote SOAP server is running, a green icon appears next to the SOAP server, as shown in the following example:

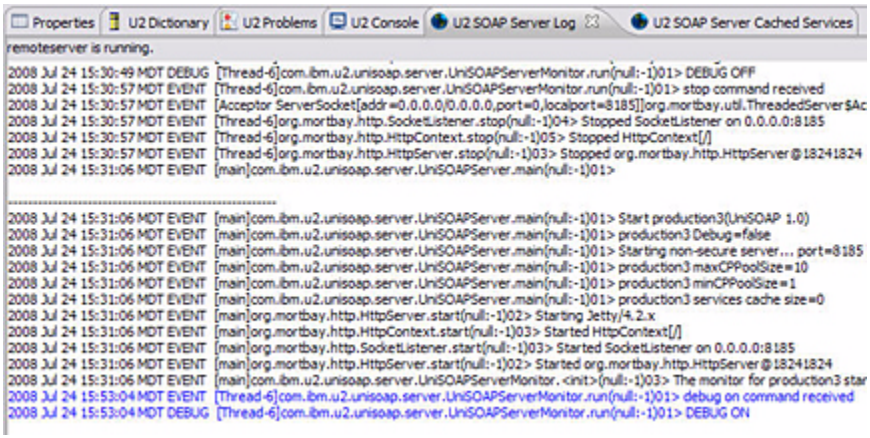


Turn on Logging

Right-click the remote SOAP server, then click **Turn on SOAP Server Debug** to start logging, as shown in the following example:



U2 Web Services Developer displays the log from the remote server as shown in the following example:



Stop the Remote SOAP Server

To stop the remote SOAP server, right-click the remote server you want to stop, then click **Stop SOAP server**.

***Note**: You cannot start the remote SOAP server from U2 Web Service Developer.*