



IBM

Basic Developer Toolkit

Version 10.3
February, 2009

IBM Corporation
555 Bailey Avenue
San Jose, CA 95141

Licensed Materials – Property of IBM

© Copyright International Business Machines Corporation 2008, 2009. All rights reserved.

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

US GOVERNMENT USERS RESTRICTED RIGHTS

Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of Contents

Preface

Chapter 1 Installing the Basic Developer Toolkit and Administering Servers

Installation Process	1-3
Administering Servers	1-9
Registering Servers	1-9
Editing Server Information	1-13
Managing Server Connections	1-14
Connecting to a Server	1-15
Disconnecting from a Server	1-16

Chapter 2 The Basic Developer Tool U2 Perspective

The U2 Resource View	2-3
Filters	2-3
Comparing Source Code	2-8
Setting UniVerse BASIC Program Options	2-13
The Basic Program Editor	2-14
Syntax Highlighting.	2-14
Code Assist	2-16
Collapsing Blocks of Text	2-22
The Outline View	2-23
Code Templates	2-24
Refactoring	2-26
Dynamic Array Editor	2-29
XML Editor	2-29

Chapter 3 Debugging a UniVerse BASIC Program

The Debug Perspective	3-6
---------------------------------	-----

Appendix A Default Symbols and Words

Default Symbols	A-2
Default Words	A-4

Preface

This manual describes the Basic Developer Toolkit. This tool enables you to write UniVerse BASIC code using a number of views and wizards, in addition to advanced editing and debugging support. Features include syntax highlighting, code assistance, a built-in compiler, and other tools.

The Basic Developer Tool runs on a client machine and accesses resource repositories and runtime environments on the UniVerse server.

Installing the Basic Developer Toolkit and Administering Servers

Installation Process	1 -3
Administering Servers	1 -9
Registering Servers	1 -9
Editing Server Information	1 -13
Managing Server Connections	1 -14
Connecting to a Server	1 -15
Disconnecting from a Server.	1 -16

This chapter describes how to install the Basic Developer Toolkit.

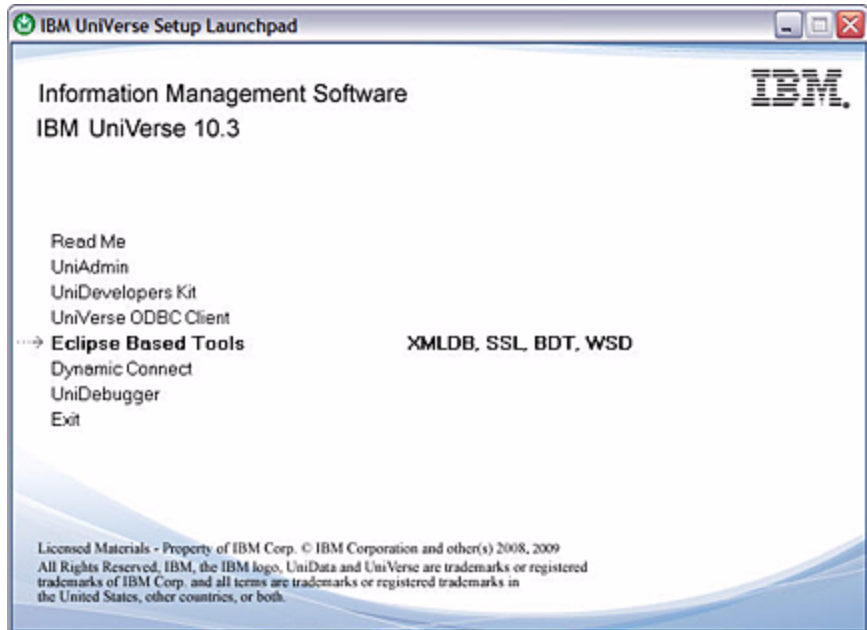
Installation Process

1. Exit Other Applications

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

2. Load the UniVerse Client CD

Place the UniVerse Client CD in your CD-ROM drive. A window similar to the following example appears:



3. Execute the Installation

From the **Setup** window, double-click **Eclipse Based Tools**. The **Welcome** window appears. Click **Next** to continue with the installation.

4. Review License Agreement

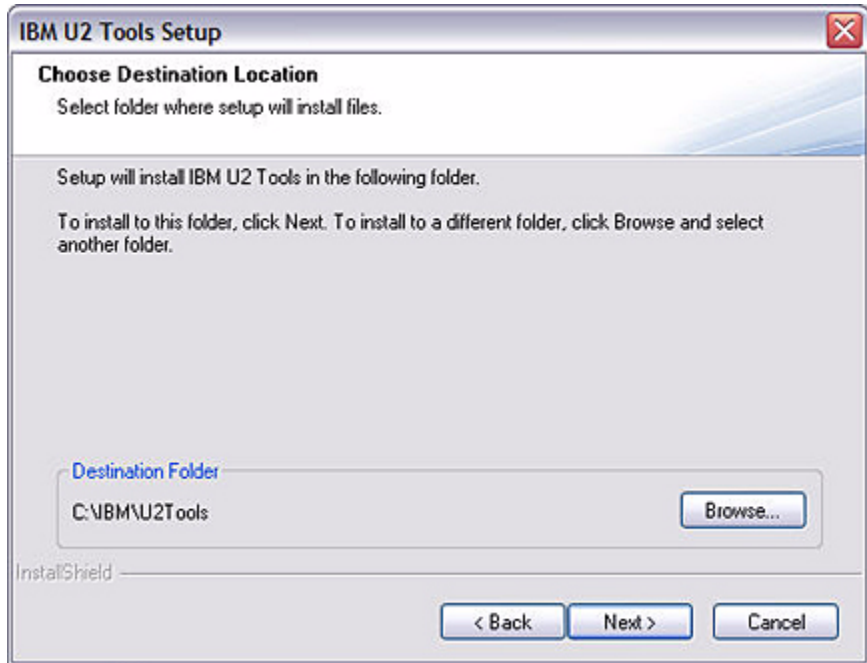
The **License Agreement** window appears, as shown in the following example:



If you agree with the terms of the license agreement, choose **I accept the terms of the license agreement**. If you do not agree with the terms of the license agreement, select **I do not accept the terms of the license agreement**. If you choose this option, you will exit the installation.

5. Choose Destination Folder

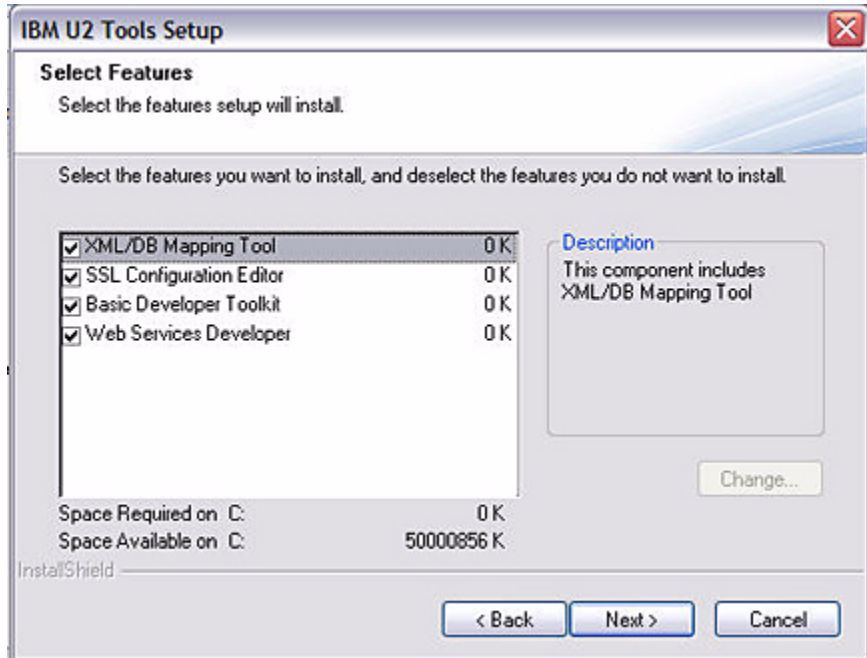
Determine where you want to install the Basic Developer Toolkit. By default, it is installed in \IBM\U2Tools. If you want to select a different directory, click **Browse**. The following example illustrates the **Choose Destination Location** dialog box:



Click **Back** to return to the previous screen, **Next** to continue with the installation, or **Cancel** to exit the installation without saving changes.

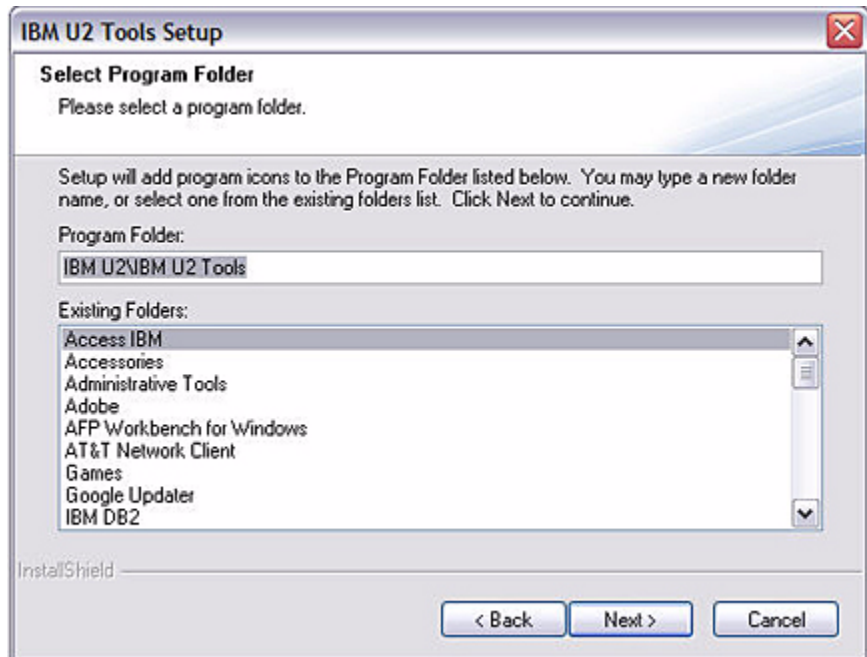
6. Select Features

Select which of the Eclipse-based tools you want to install. Make sure you select the **Basic Developer Toolkit** check box. The following example illustrates the **Select Features** dialog box:



7. Select Program Folder

Determine the name of the program folder where you want to install the Basic Developer Toolkit. The following example illustrates the **Select Program Folder** dialog box:

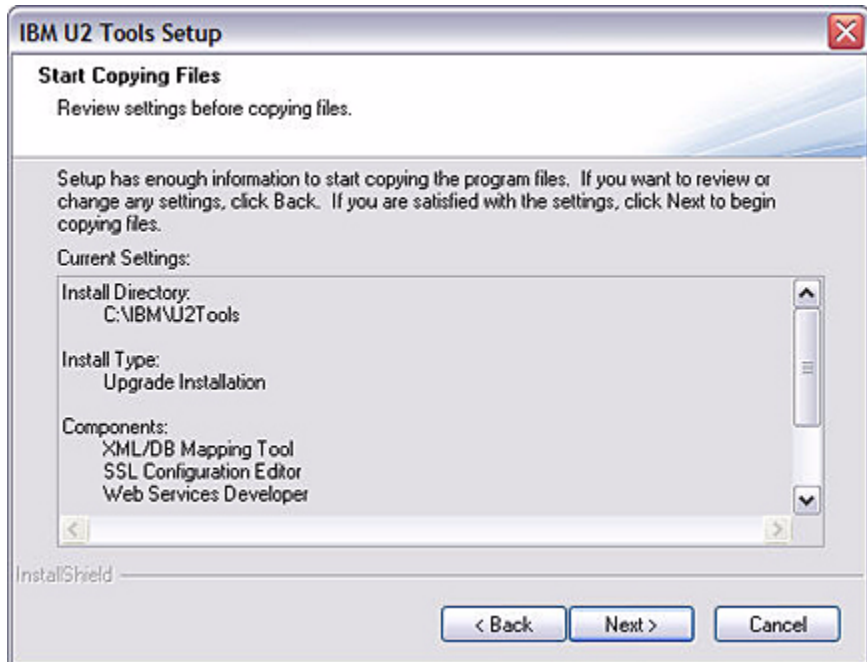


By default, the installation process installs the Basic Developer Toolkit in the IBM U2\IBM U2 Tools folder. If you want to install it in a different folder, enter the name of the folder in the **Program Folder** box, or select the folder from the **Existing Folders** list.

Click **Back** to return to the previous screen, **Next** to continue with the installation, or **Cancel** to exit the installation without saving changes.

8. Copy Files

UniVerse now has enough information to begin copying files. A dialog box similar to the following example appears:



If you want to change any of the settings, click **Back**. Click **Next** to continue with the installation process, or click **Cancel** to exit the installation program.

Administering Servers

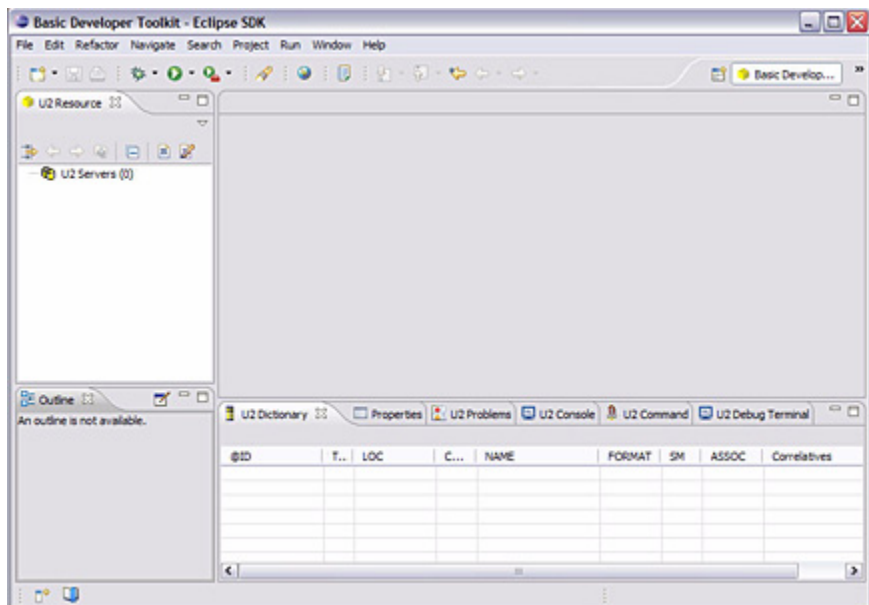
You must register each server that you want to access through the BASIC Developer Toolkit. The server must be running UniVerse 10.3 or later.

Registering Servers

Complete the following steps to register servers for the BASIC Developer Toolkit.

1. Start the Basic Developer Toolkit

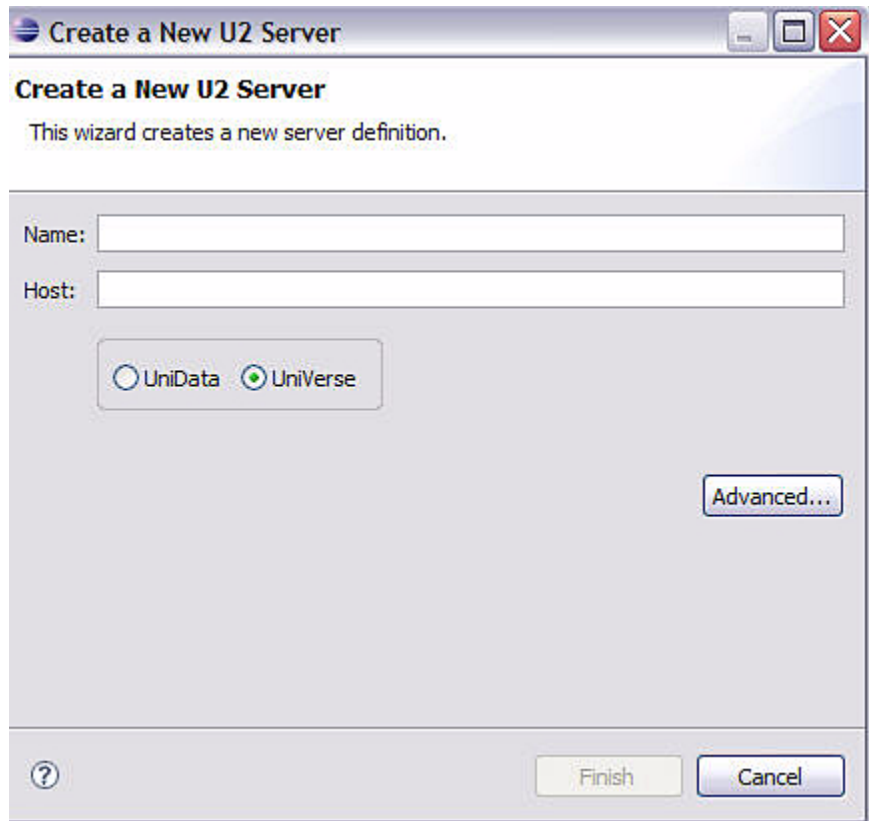
From the **Start** menu, select **All Programs**, then select the folder where the Basic Developer Toolkit resides, and then click **Basic Developer Toolkit**. A window similar to the following example appears:



2. Define Server to Register

To define a server to register, using the right mouse button (right-click) **U2Servers**, and then click **New U2 Server**.

A dialog box similar to the following appears:



The screenshot shows a Windows-style dialog box titled "Create a New U2 Server". The title bar includes standard minimize, maximize, and close buttons. The main content area has a subtitle "Create a New U2 Server" and a description "This wizard creates a new server definition." Below this, there are two text input fields: "Name:" and "Host:". Under the "Host:" field, there are two radio buttons: "UniData" (unselected) and "UniVerse" (selected). To the right of these radio buttons is an "Advanced..." button. At the bottom of the dialog, there is a question mark icon on the left, and "Finish" and "Cancel" buttons on the right.

Enter the name of the server you want to register in the **Server Name** box. Enter the host name or IP address in the **Host** box.

3. Choose Database

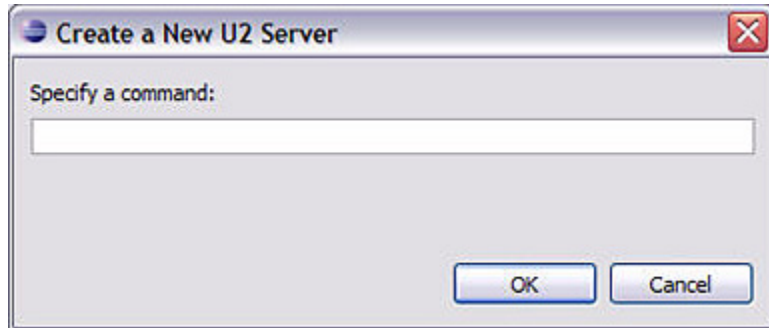
Select the database against which you are using the Basic Developer Toolkit. The database is either UniData or UniVerse.

4. Choose Advanced Options

To define the following options, click **Advanced**:

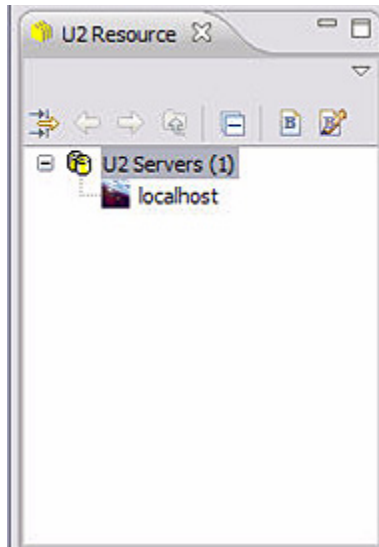
- **Protocol Type** – In the **Protocol Type** box, make sure the type of communication you are using to the server is Default or TCP/IP.
- **RPC Port Number** – In the **RPC Port #** box, enter the port number of the UniRPC server running on the host. The default port number is 31438.
- **RPC Service Name** – In the **RPC Service Name** box, enter the name of the RPC service on your system. For UniVerse, this is normally uves.
- **Login Account** – In the **Login Account** box, enter the name of the account to which you want to log on when accessing UniVerse.

- **Commands to Execute** – If you want to execute a Retrieve command, a saved paragraph, or a globally cataloged program, click **Add**. A dialog box similar to the following example appears:



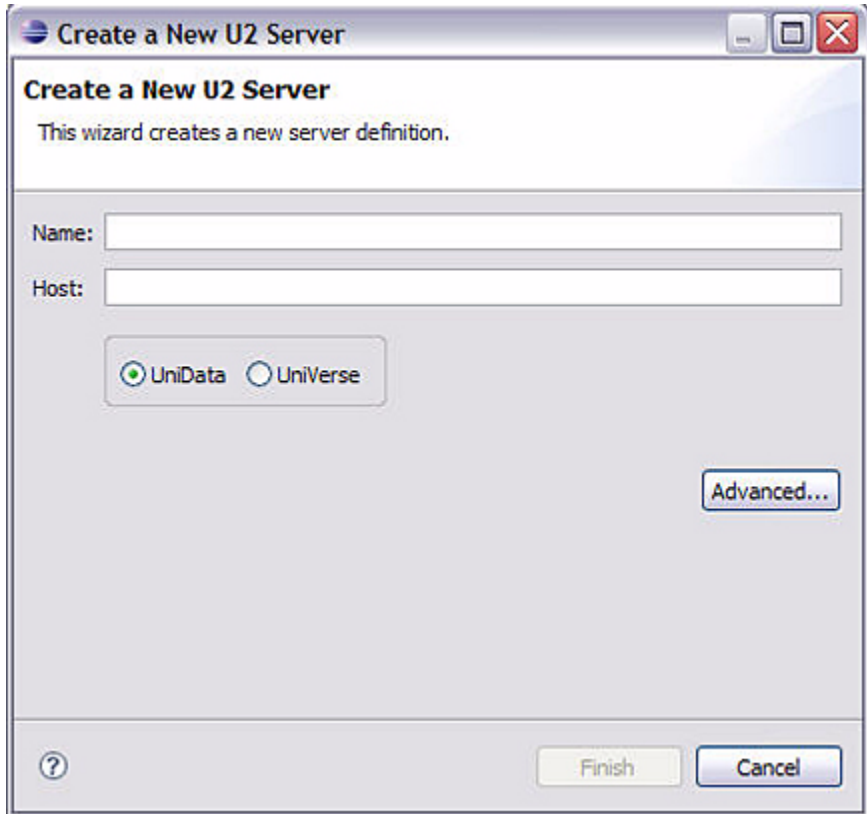
Enter the Retrieve command, save paragraph name, or globally cataloged program name in the **Specify a Command** box, then click **OK**.

Click **Finish** to return to the **Create a New U2 Server** dialog box, then click **Finish** to register the server. The name of the server appears in the **U2 Resource** section of the Basic Developer Toolkit, as shown in the following example:



Editing Server Information

If you want to change information you previously entered about a registered server, right-click on the server for which you want to edit information, and then click **Properties**. A dialog box similar to the following appears:

A screenshot of a Windows-style dialog box titled "Create a New U2 Server". The title bar includes standard minimize, maximize, and close buttons. The main area of the dialog has a light blue header with the title and a subtitle: "This wizard creates a new server definition." Below this, there are two text input fields labeled "Name:" and "Host:". Under the "Host:" field, there are two radio buttons: "UniData" (which is selected, indicated by a green dot) and "UniVerse". To the right of these options is a button labeled "Advanced...". At the bottom of the dialog, there is a question mark icon on the left, and two buttons labeled "Finish" and "Cancel" on the right.

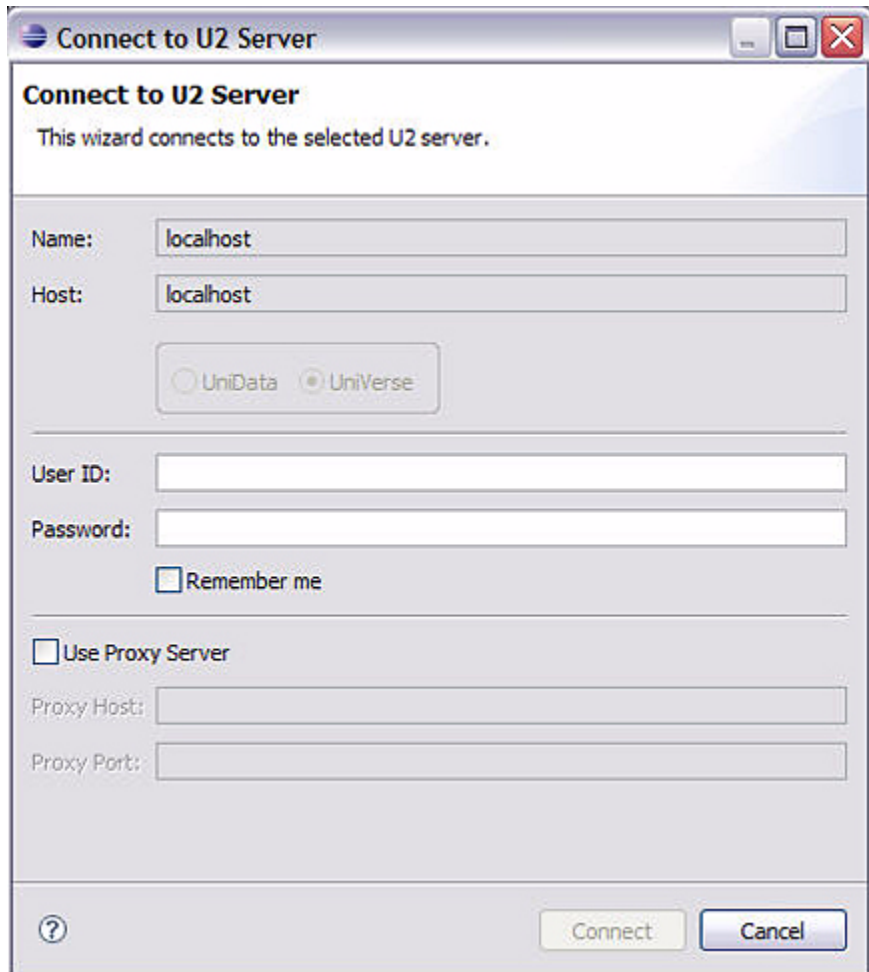
Edit the information you want to change. Click **Finish** to save the changes, or click **Cancel** to exit the dialog box without saving changes.

Managing Server Connections

This section describes how to connect and disconnect from a UniVerse server.

Connecting to a Server

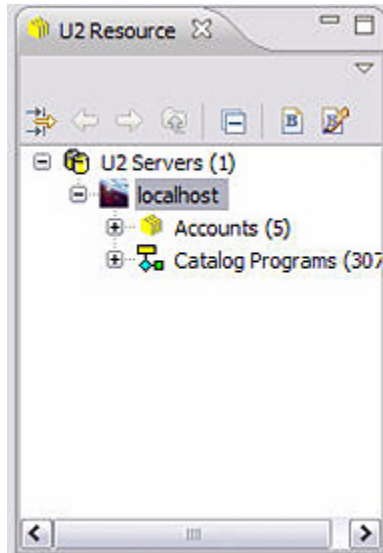
To connect to a server, double-click the server to which you want to connect, or right-click the server and then click **Connect**. A dialog box similar to the following example appears:



The image shows a Windows-style dialog box titled "Connect to U2 Server". The title bar includes standard minimize, maximize, and close buttons. The main content area has a header section with the title "Connect to U2 Server" and a subtitle "This wizard connects to the selected U2 server." Below this, there are two text input fields: "Name:" and "Host:", both containing the text "localhost". Under these fields is a group box containing two radio buttons: "UniData" (which is unselected) and "UniVerse" (which is selected). Below the group box are two more text input fields: "User ID:" and "Password:". Below the "Password:" field is a checkbox labeled "Remember me", which is currently unchecked. Further down is another checkbox labeled "Use Proxy Server", also unchecked. Below this checkbox are two more text input fields: "Proxy Host:" and "Proxy Port:". At the bottom of the dialog, there is a question mark icon on the left, and two buttons on the right: "Connect" and "Cancel".

Enter the login name of the user you are connecting as in the **User ID** box. Tab to the **Password** box and enter the password for the user.

After you connect to the server, the **U2 Resource** section of the Basic Developer Toolkit displays the Accounts and Cataloged programs on the server to which you are connected, as shown in the following example:



Disconnecting from a Server

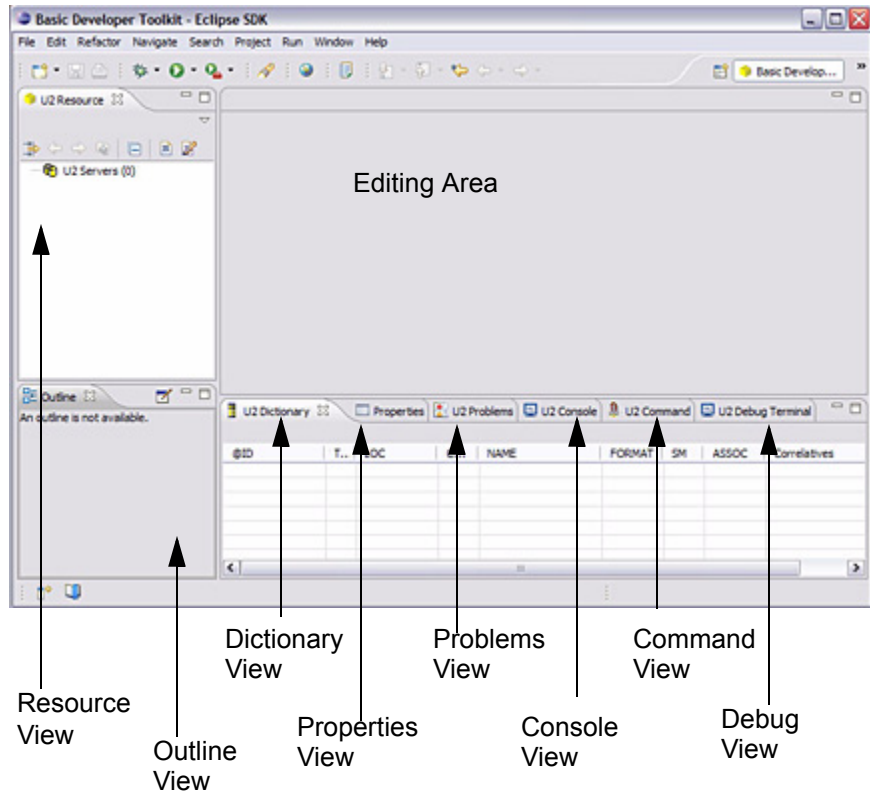
To disconnect from a server, right-click on the server from which you want to disconnect, and then click **Disconnect**.

The Basic Developer Tool U2 Perspective

The U2 Resource View	2 -3
Filters	2 -3
Comparing Source Code	2 -8
Setting UniVerse BASIC Program Options	2 -13
The Basic Program Editor	2 -14
Syntax Highlighting	2 -14
Code Assist	2 -16
Collapsing Blocks of Text	2 -22
The Outline View	2 -23
Code Templates.	2 -24
Refactoring	2 -26
Dynamic Array Editor.	2 -29
XML Editor	2 -29

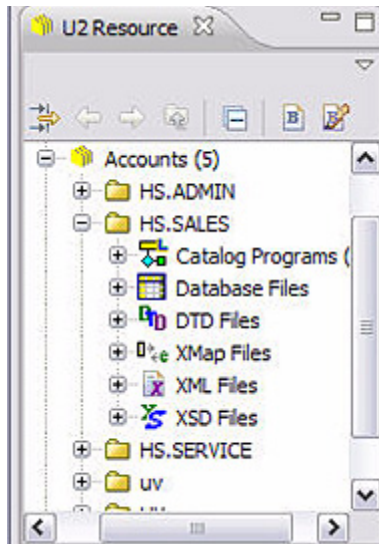
This chapter describes the Basic Developer Toolkit perspective.

The following example illustrates the default Basic Developer Toolkit perspective:



The U2 Resource View

The U2 Resource View contains information about each UniVerse account of the server to which you are connected. This information includes accounts, data files, dictionary files, UniVerse BASIC programs, XML/DB mapping files, and cataloged programs. The following example illustrates the U2 Resource information for the HS.SALES account:



Filters

You can establish filters to filter out information you do not want the Basic Developer Toolkit to display. There are two types of filters, Resource-type filters and pattern filters.

Resource-type Filters

Resource-type filters include server filters, account filters, and file filters.

Server Filter

Use the server filter to exclude UniVerse servers from being displayed in the Basic Developer Toolkit. By default, the server filter is empty.

Account Filter

Use the account filter to exclude UniVerse accounts from being displayed in the Basic Developer Toolkit. By default, the Basic Developer Toolkit filters out UniVerse system accounts using the following pattern:

Master*, sys*, bin_*

File Filter

Use the file filter to exclude UniVerse files from being displayed in the Basic Developer Toolkit. By default, UniVerse excludes the following files:

- CTLG
- SAVEDLISTS
- &HOLD&
- &PH&
- &XML&
- savedlists
- &MAP&
- &report&
- AE-related files
- CTLGTB
- ENGLISH.MSG
- ERRMSG
- HELP.FILE
- MENUFILE
- REP_RECV_LOG
- REP_RECV_REC
- VOC
- &DEBUG&

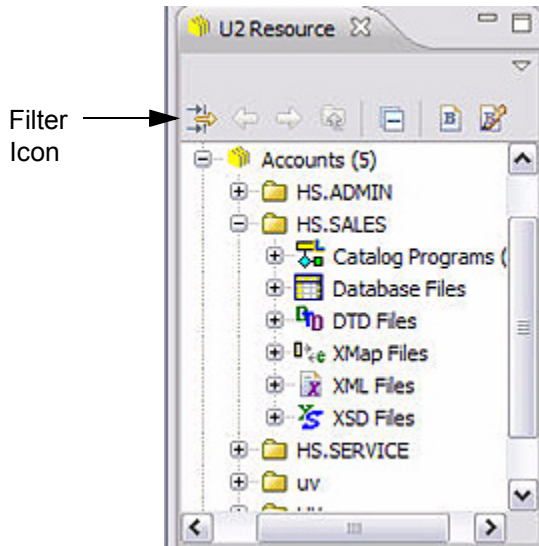
- &MAP&
- voc
- privilege
- &REPORT&
- &SCREEN&
- UV.ACCOUNT
- APP.PROGS
- BASIC.HELP
- catdir
- GLOBAL.CATDIR
- NLS-related files
- DICT. files
- UNIVERSE. files
- UV_ files
- UV. files
- SYS. files
- REVISE. files

Pattern Filters

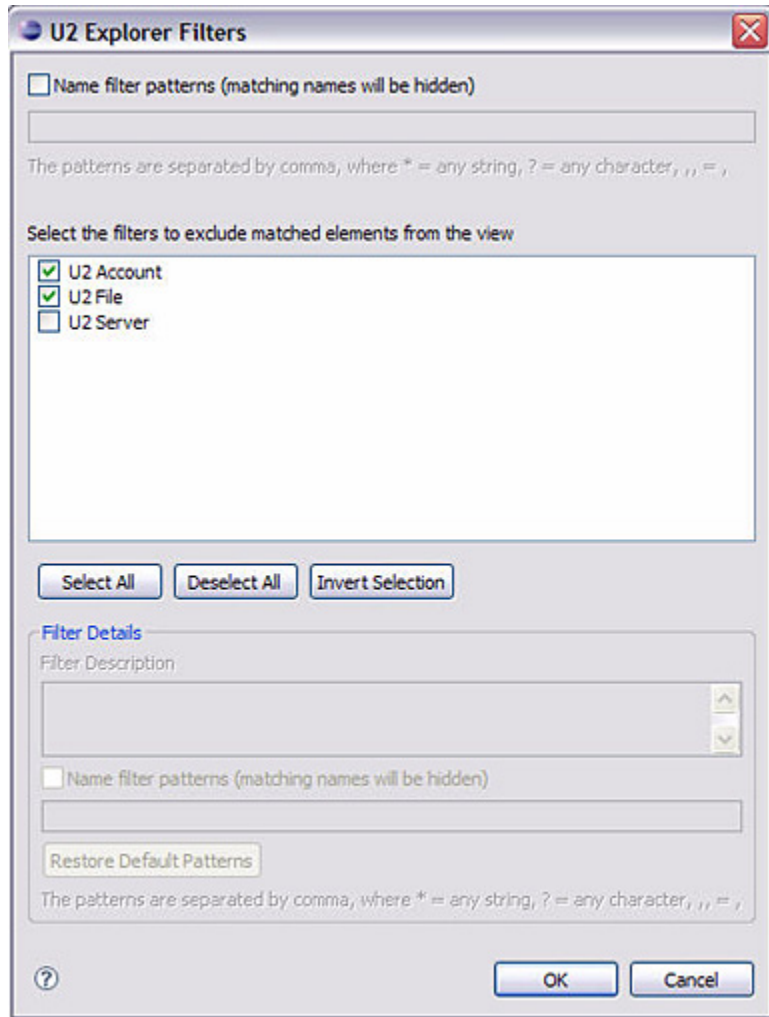
Use a pattern filter to exclude any name defined by the pattern you specify from being displayed in the Basic Developer Toolkit.

Creating a Filter

To create a filter, click the Filter icon in the U2 Resource View, as shown in the following example:



The **U2 Explorer Filters** dialog box appears, as shown in the following example:



If you want to create a pattern filter, select the **Name filter patterns** check box. Separate each pattern with a comma (.). Pattern matching characters are:

- * = any string
- ? = any character
- ,, (2 commas) = , (comma)

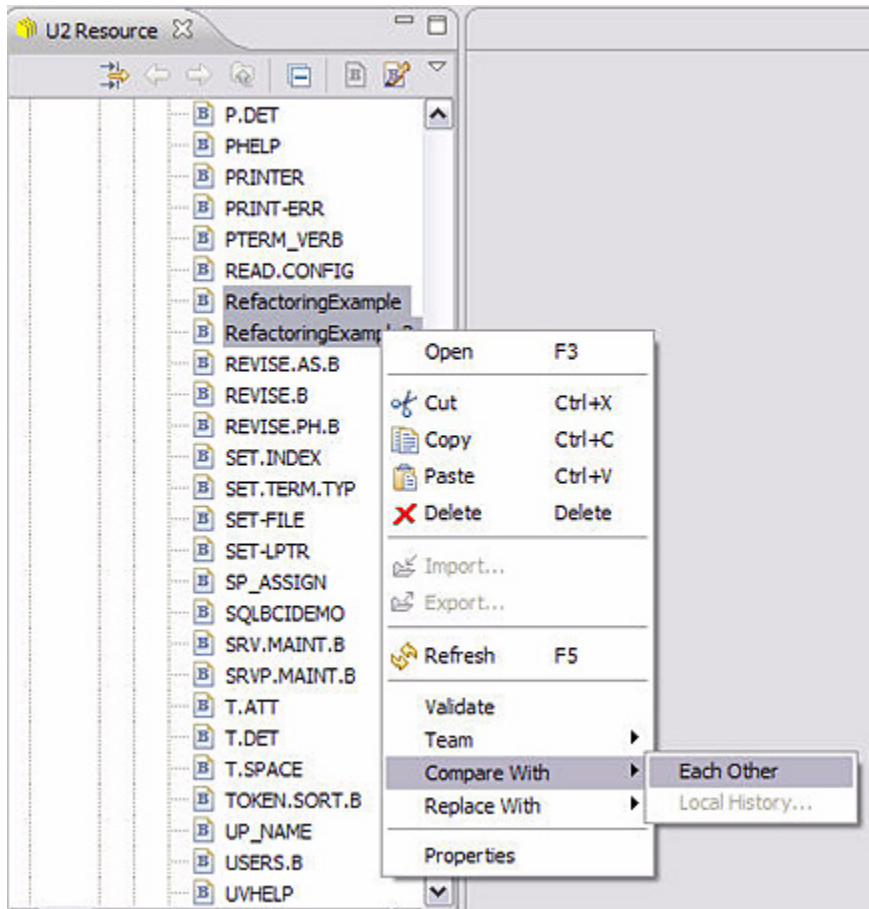
Select the type of Resource filter you want to create in the **Select the filters to exclude matched elements from the viewarea**. You can also click **Select All** to select all of the filters, **Deselect All** to clear all of the filters, or **Invert Selection**.

In the **Filter Details** dialog box, enter the pattern matching details for the type of filter you selected.

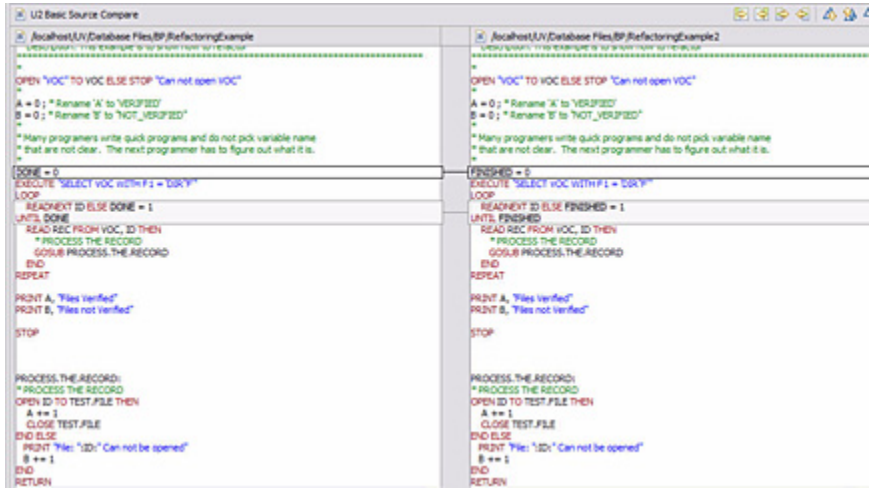
Comparing Source Code

You can compare two UniVerse BASIC programs with each other to display the differences in the programs.

Select the first program you want to compare in the account and file where it resides. Holding down the CTRL key, select the second program with which you want to compare the first. Using the right mouse key (right-click), select **Compare With**, the select **Each Other**, as shown in the following example:

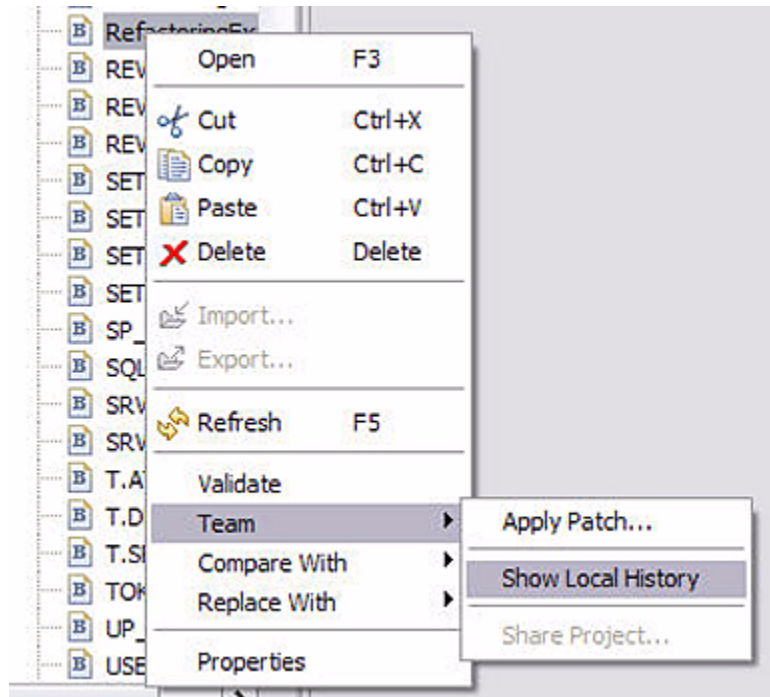


In the following example, the RefactoringExample2 program has the DONE variable changed to FINISHED. The first program you select appears in the left frame, while the second program you select appears in the right frame. The Basic Developer Toolkit highlights the differences in the programs:

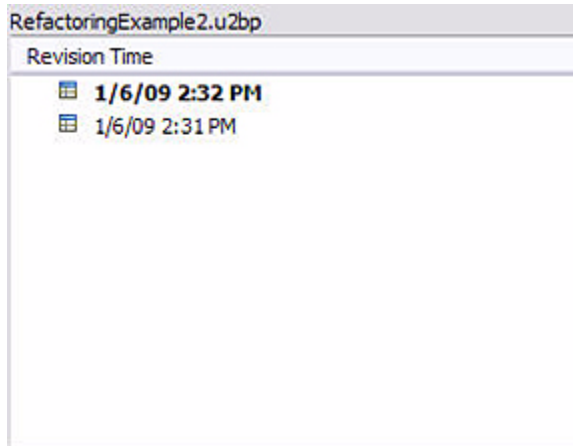


Comparing Local History

You can compare the history of changes to a program made on your local machine. To compare local history, right-click the program for which you want to view history, then select **Team**, then click **Show Local History**, as shown in the following example:



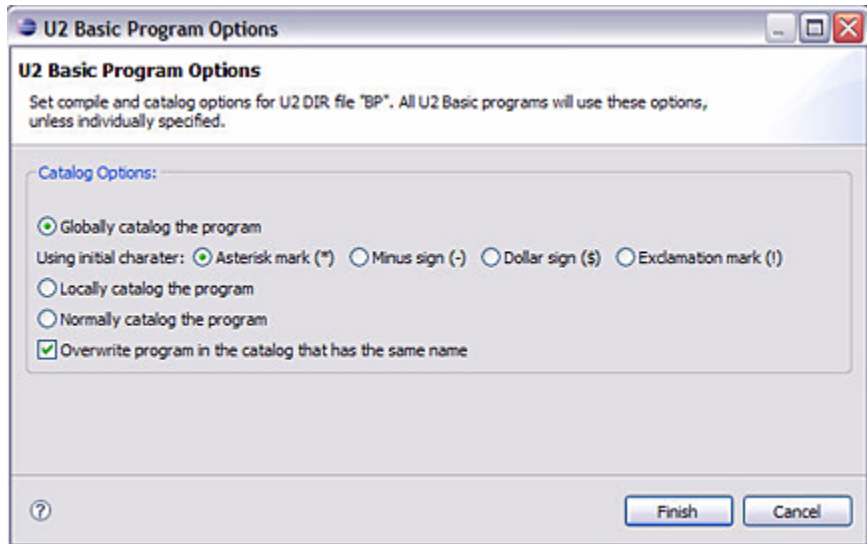
The Basic Developer Toolkit shows the local history for the program you selected, as shown in the following example:



You can compare changes made to the program by select the two revision times you want to compare, then follow the steps in [“Comparing Source Code”](#) on page 8.

Setting UniVerse BASIC Program Options

You can specify compilation and cataloging options for UniVerse BASIC programs at the account-level or the program-level. To set program options, right-click the account or directory file for which you want to set the options, then click **U2 Basic Program Options**. A dialog box similar to the following example appears:



Select the catalog options for the directory or account, then click **Finish**.

***Note:** Program options set at the directory-level override program options set at the account-level.*

The Basic Program Editor

The Basic Program Editor provides several options to assist you in creating or editing UniVerse BASIC programs, including:

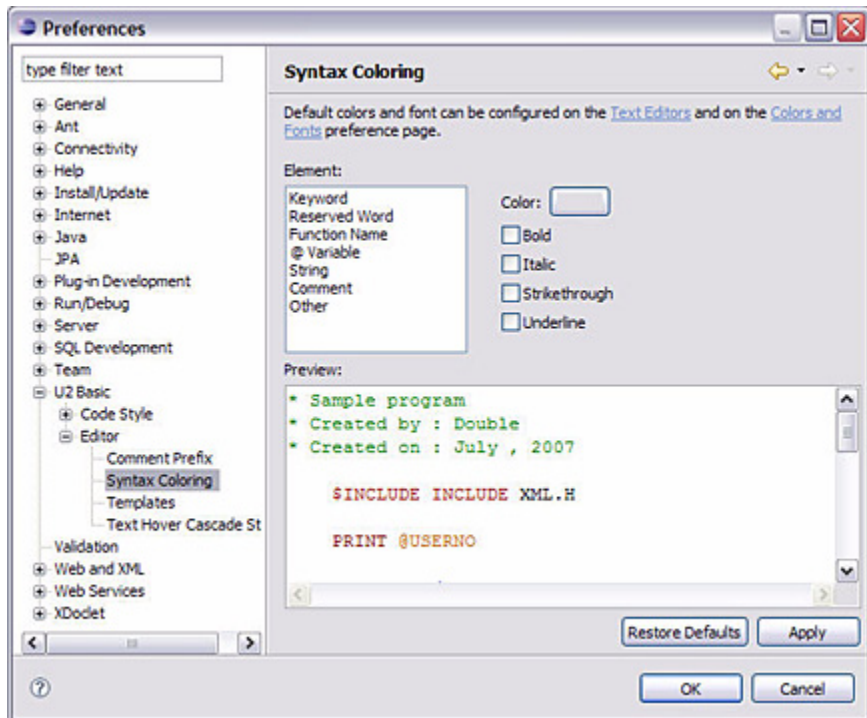
- [Syntax Highlighting](#)
- Code Assist

Syntax Highlighting

The Basic Developer Toolkit recognizes UniVerse BASIC syntax, and provides syntax highlighting. You can configure the colors used and formatting for the code. The Basic Developer Toolkit highlights syntax error marks after each unsuccessful compilation.

Customizing Syntax Colors

To customize the colors for syntax highlighting, click **Window**, then click **Preferences**, then click **U2 Basic**, expand **Editor**, then click **Syntax Coloring**. A dialog box similar to the following example appears:



You can customize the coloring for the following elements:

- Keyword
- Reserved word
- Function Name
- @ Variable
- String
- Comment
- Other

Code Assist

Code Assist uses code templates and helps in avoiding syntax errors. Code assist checks the following elements in addition to the code template:

- Verbs
- Keywords
- Functions
- @ Variables
- \$ Variables
- Defined labels
- Defined variables
- Defined user functions

The Basic Developer Toolkit uses as set of default verbs, keywords, functions, @ variables and \$ variables, defined in Appendix A, “[Default Symbols and Words.](#)” You can override these defaults by creating the XTOOLSUB.DATA file in the *universe_home/include* directory. This file contains two sections, one for symbols and one for words.

Specifying Symbols

The symbols section should start with “*symbols” and each line of the section should have the following format:

{“symbol_name”, symbol_type, 0}

The symbol_type should be one of the following:

- Vvariable – @ Variable (read/write)
- Vrvariable – @ Variable (read only)
- Vprequate – Pre-defined equate statement

The following example illustrates the symbols section of the file:

```
*
*symbols
*
{ "@SYSTEM.RETURN.CODE", Vavvariable,      0 }
{ "@STDFIL",             Vavvariable,      0 }
{ "@SELECTED",           Vavvariable,      0 }
{ "@IM",                 Vprequate,        0, "Add: more ITM" }
{ "@FM",                 Vprequate,        0 }
{ "@AM",                 Vprequate,        0 }
{ "@VM",                 Vprequate,        0 }
.
.
.
```

Specifying Words

The words section should start with “*words” and each line of the section should have the following format:

```
{“name”, type, num_of_args, max_num_of_args, optional_arg_flag,
left_value_arg_list, optional_description}
```

The following table describes valid values for the type variable:

Type	Description
Lunknown	Type unknown
Lbflag	Statement that starts at the line.
Lbfunc	Function style statement (CONVERT, DELETE, REMOVE)
Lbos	Beginning of line statement
Lelse	Special case: ‘ELSE’
Lend	Special case: ‘END’
Lflag	Check keyflag
Lfunct	Check for following ‘(‘
Lloc	Special case: ‘LOCATE’
Llock	Special case: ‘LOCKED’ (also checks keyflag)
Lloop	Special case: ‘UNTIL’ & ‘WHILE’

words Section Types

Type	Description
Lnext	Special case: 'NEXT'
Lsub	Special case: 'SUBROUTINE'
Lthen	Special case: 'THEN'
Lget	Special case: 'GET ('
Lseek	Special case: 'SEEK ('
Lprntr	Special case: 'PRINTERR'
Lexec	Special case: 'PERFORM'
Lstat	Special case: 'STATUS'
Error	Special case: '(ON) ERROR
Lbig	Special case: '\$DEFINE, \$UNDEFINE
Lcdir	Special case: '\$ELSE, \$ENDIF

words Section Types (continued)

The code assist feature uses the *num_of_args*, *max_num_of_args*, *optional_arg_flag*, and *left_value_arg_list* for functions only.

The following example illustrates the *words portion of the XTOOLSUB.DATA file:

```

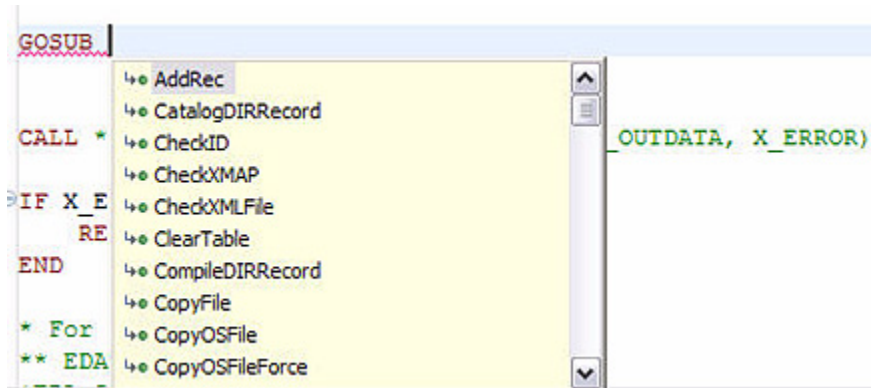
*
*words
*
    { "$DEFINE",          Lbos, 0, 0, 0, 0}
    { "$ELSE",           Lcdir, 0, 0, 0, 0}
    { "$ENDIF",          Lcdir, 0, 0, 0, 0}
    { "$F",              0, 0, 0, 0, 0}
    { "$FALSE",          0, 0, 0, 0, 0}
    { "$IFDEF",          Lbig, 0, 0, 0, 0}
    { "$IFNDEF",         Lbig, 0, 0, 0, 0}
    { "$OPTIONS",        Lbos, 0, 0, 0, 0}
.
.
.
```

Using Code Assist with GOSUB

If you enter a GOSUB or GOTO statement, The Basic Developer Toolkit displays all the labels contained in the program. Enter:

GOSUB (CTRL+SPACE)

The Basic Developer Toolkit displays all labels with the program, as shown in the following example:



Select the correct subroutine from the list.

Using Code Assist with CALL

If you enter a CALL statement followed by a (CTRL + space), the Basic Developer Toolkit displays all globally and locally cataloged programs in alphabetical order, as shown in the following example:

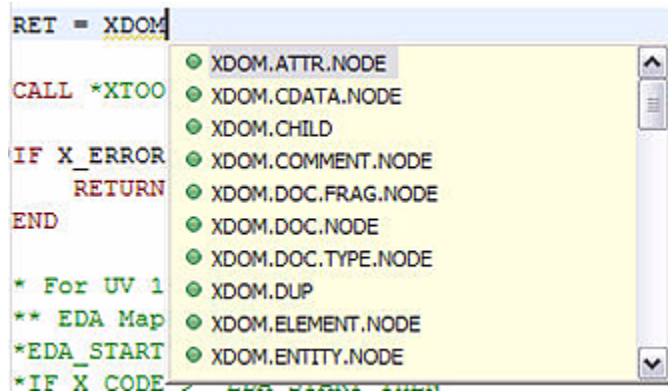


Using Code Assist with Functions

When you enter the name of a UniVerse function, the Basic Developer Toolkit displays details information for the function. In the next example, the following code was entered:

```
RET = XDOM(CTRL+SPACE)
```

The Basic Developer Toolkit displays all functions beginning with “XDOM” as shown in the following example:



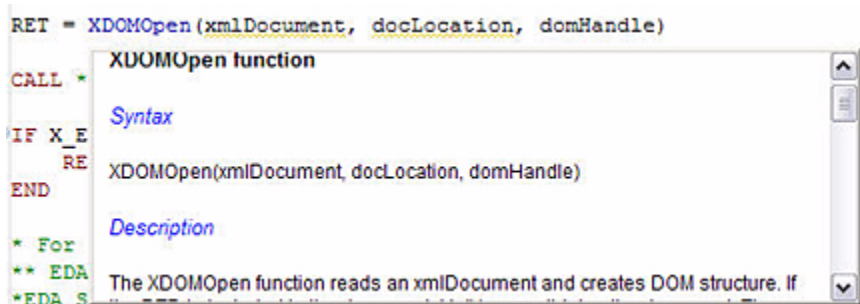
When you select the function, the Basic Developer Toolkit inserts the function as a template, as shown in the following example:

```
RET = XDOMOpen(xmlDocument, docLocation, domHandle)
```

You can enter the tab key to enter your values for the function parameters.

Viewing Syntax

You can hover over a UniVerse BASIC function or statement to view the syntax. After the syntax appears, click the F2 to keep the syntax on the screen. The following example illustrates the syntax for the XDOMOpen function:



Collapsing Blocks of Text

You can collapse certain blocks of text within the program. Consider the following LOOP statement:

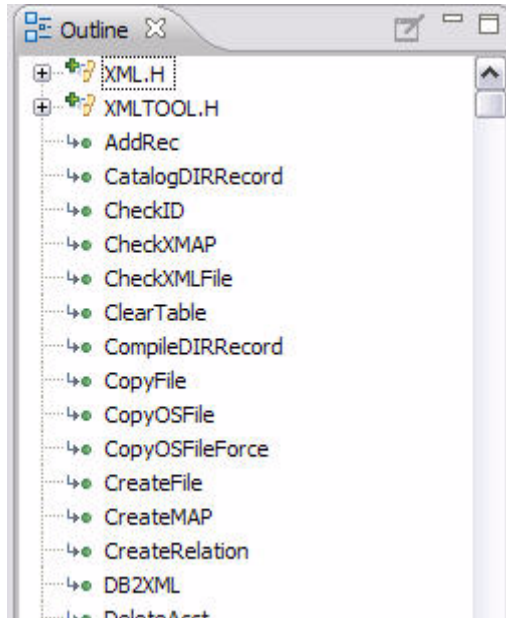
```
- LOOP
    READNEXT ID ELSE DONE = 1
UNTIL DONE
    READ REC FROM VOC, ID THEN
        * PROCESS THE RECORD
        GOSUB PROCESS.THE.RECORD
    END
REPEAT
```

If you click the “-” icon, the Basic Developer Toolkit collapses the statement. After the statement is collapsed, you can hover over the “+” to view the statement, as shown in the following example:

```
+ LOOP
    READNEXT ID ELSE DONE = 1
UNTIL DONE
    READ REC FROM VOC, ID THEN
        * PROCESS THE RECORD
        GOSUB PROCESS.THE.RECORD
    END
REPEAT
```

The Outline View

The Basic Developer Toolkit displays all procedures and labels that appear in the source code you specify. With this outline, you can easily jump to appropriate reference in the source code. The following example illustrates an outline view:



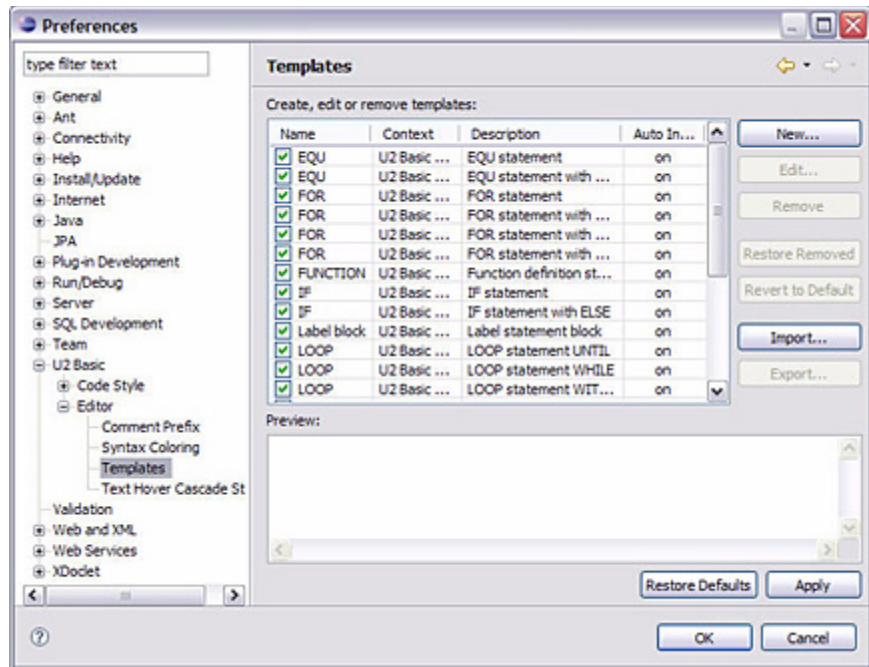
The Basic Developer Toolkit uses the following icons:

	Included File
	Label
	Variable or user function
	Common variable
	Array

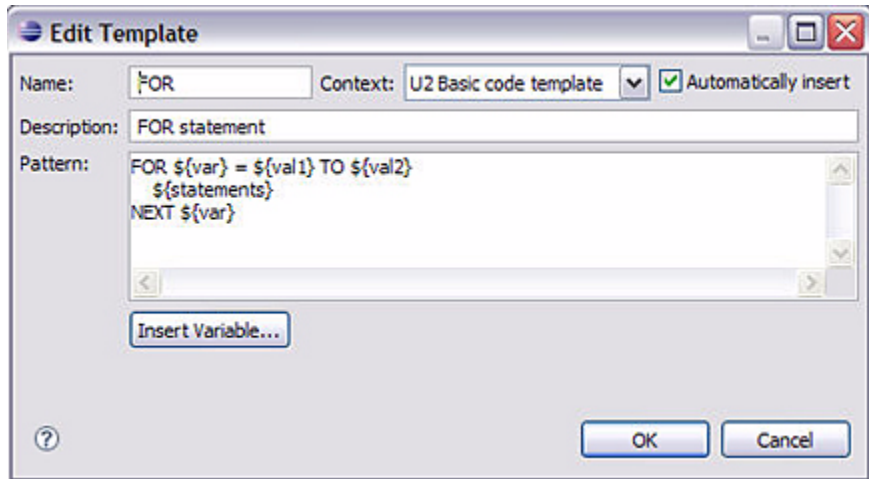
Code Templates

The Basic Developer Toolkits allows you to use code templates, which are definitions of the UniVerse BASIC syntactical constructions. You can create your own code templates to expand your own shortcuts, such as a template to insert the author and date for the source code.

From the **Window** menu, click **Preferences**, expand **U2 Basic**, expand **Editor**, then click **Templates**. A dialog box similar to the following example appears:



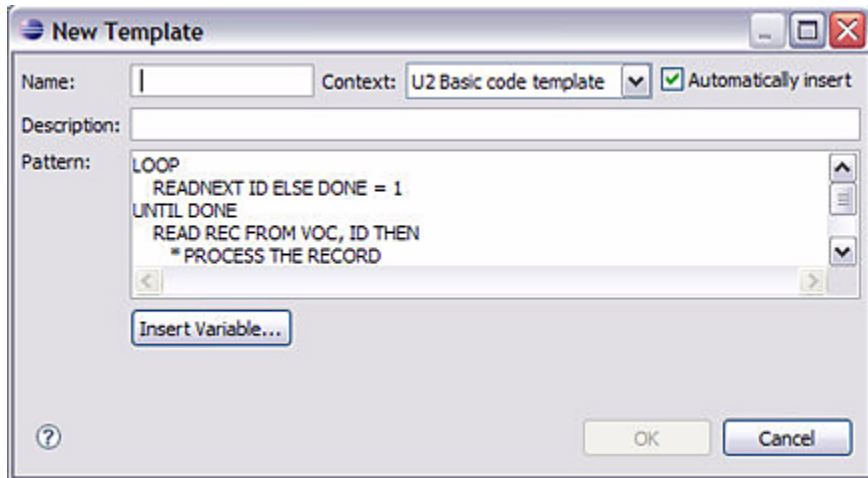
The following example illustrates the template for the FOR statement:



To create a new template, from the **Templates** dialog box, click **New**. To edit and existing template, click **Edit**. To delete a template, click **Remove**.

Creating a Template Automatically

From the Basic Developer Toolkit Editor, you can automatically create a template by marking a segment of the source code. After marking the source code, click the **U2 Basic**, then click **Copy Selection as Template**. A dialog box similar to the following example appears:

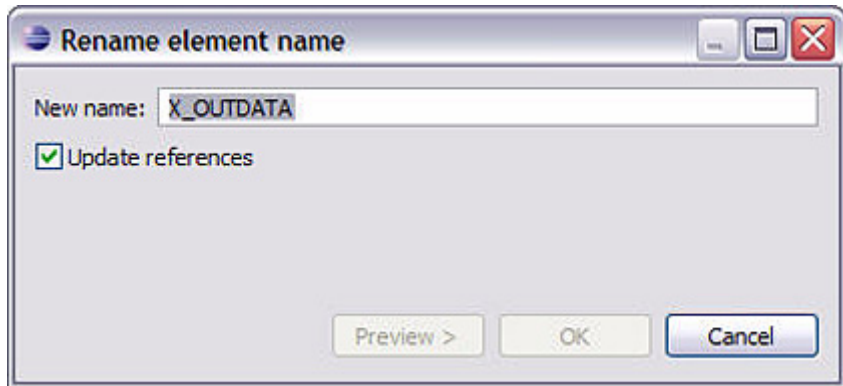


In the **Name** box, enter the name for the new template. Enter a description for the template in the **Description** box.

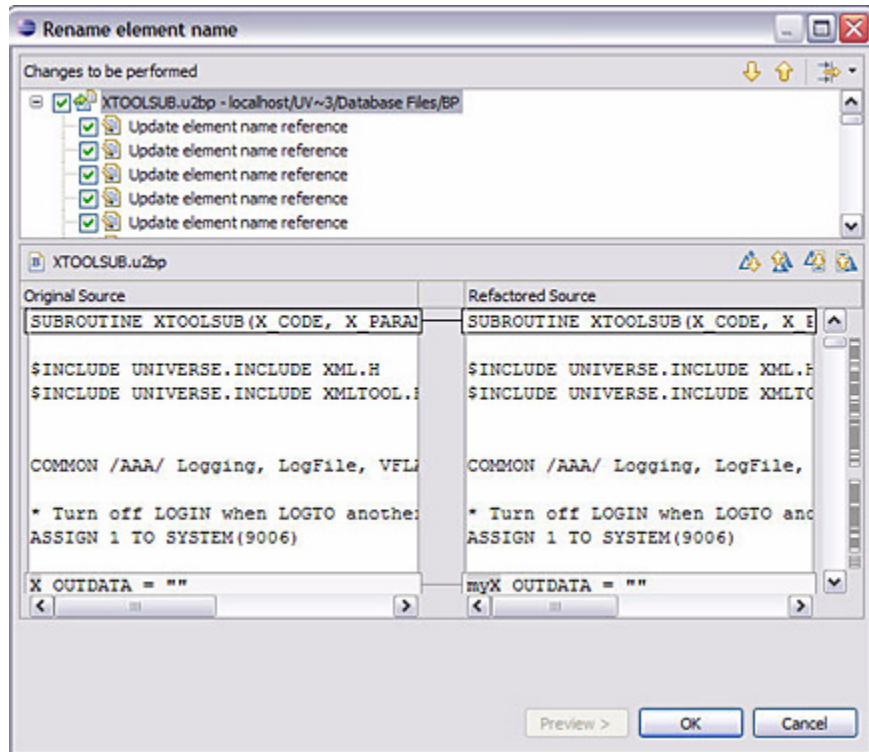
Refactoring

The Basic Developer Toolkit supports refactoring for renaming variables or labels in the current UniVerse BASIC source code. Refactoring changes a program's internal structure without modifying its external behavior or existing functionality. You usually refactor a program to improve the readability, improve extensibility, or simplify code structure.

Right-click the name of an element you want to change in your UniVerse BASIC source code, expand **Refactor**, then click **Rename**. A dialog box similar to the following example appears:



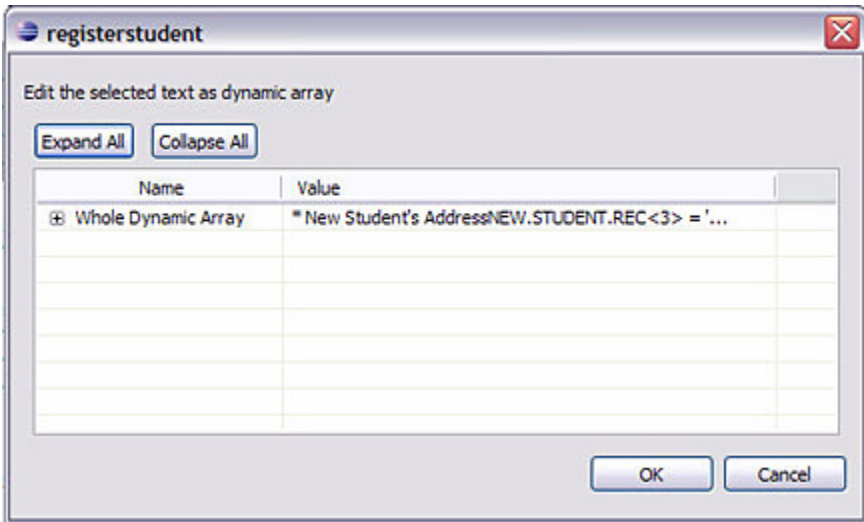
Enter the new name for the element, then click **Preview**. The Basic Developer Toolkit displays the instances of the original variable name and previews the changes, as shown in the following example:



To process the changes, click **OK**, or click **Cancel** to exit the program without changing the element name.

Dynamic Array Editor

You can edit a program using a dynamic array editor. Select the source code you would like to edit, then click the **U2 Basic** menu, then click **Edit Selection as Dynamic Array**. The dynamic array editor appears, as shown in the following example:

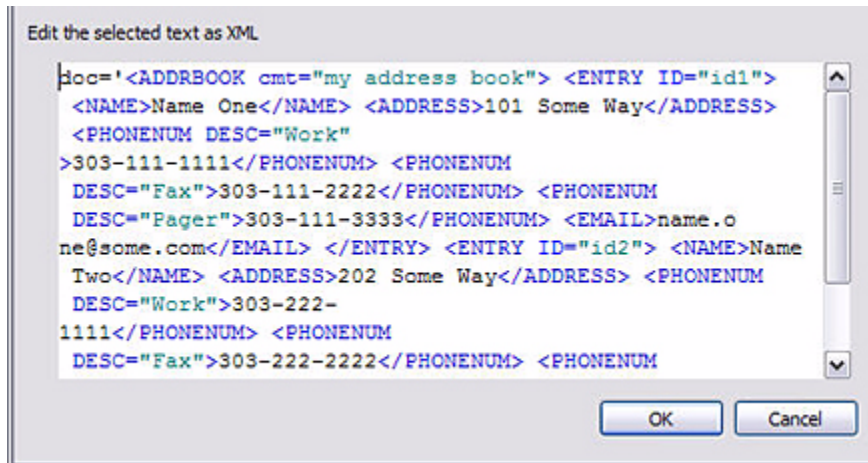


XML Editor

You can edit a program using an XML editor. Select the source code you would like to edit, as shown in the following example:

```
doc='<ADDRBOOK cmt="my address book"> <ENTRY ID="id1"> <NAME>Name
One</NAME> <ADDRESS>101 Some Way</ADDRESS> <PHONENUM DESC="Work"
>303-111-1111</PHONENUM> <PHONENUM DESC="Fax">303-111-2222</PHONENUM>
<PHONENUM DESC="Pager">303-111-3333</PHONENUM> <EMAIL>name.o
ne@some.com</EMAIL> </ENTRY> <ENTRY ID="id2"> <NAME>Name Two</NAME>
<ADDRESS>202 Some Way</ADDRESS> <PHONENUM DESC="Work">303-222-
1111</PHONENUM> <PHONENUM DESC="Fax">303-222-2222</PHONENUM> <PHONENUM
DESC="Home">303-222-3333</PHONENUM> <EMAIL>name.two@some.co
m</EMAIL> </ENTRY> </ADDRBOOK>'
```

Click the **U2 Basic** menu, then click **Edit Selection as XML**. The XML editor appears, as shown in the following example:

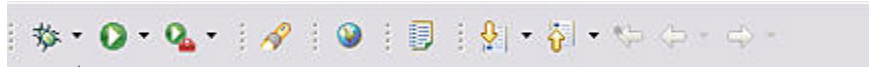


Debugging a UniVerse BASIC Program

The Debug Perspective 3 -6

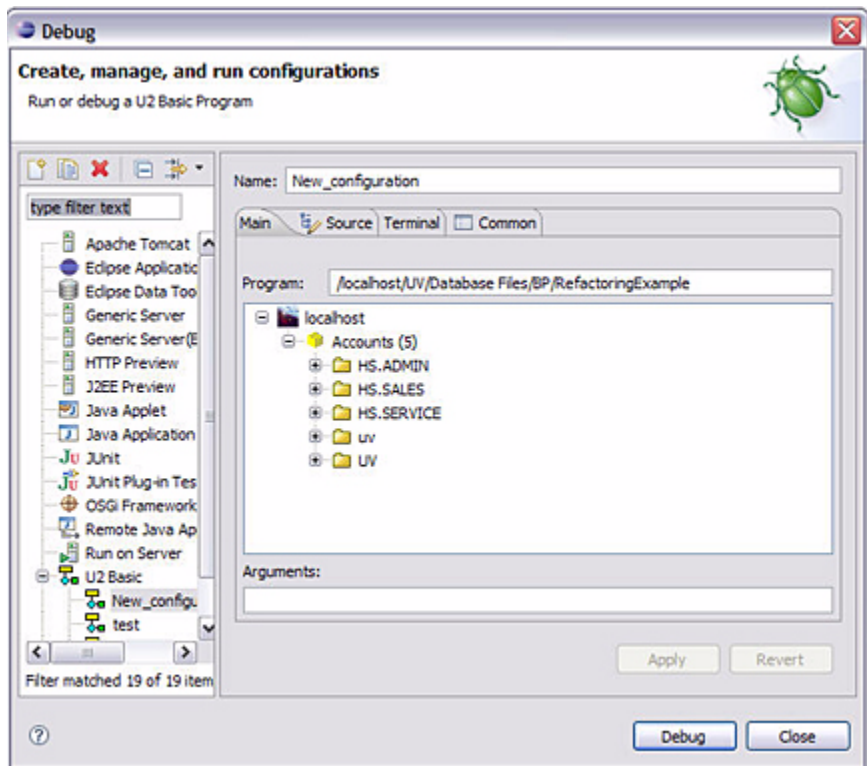
This chapter describes how to debug a UniVerse BASIC program using the Basic Developer Toolkit.

From the Basic Developer Toolkit toolbar, click the arrow next to the Debug icon, as shown in the following example:

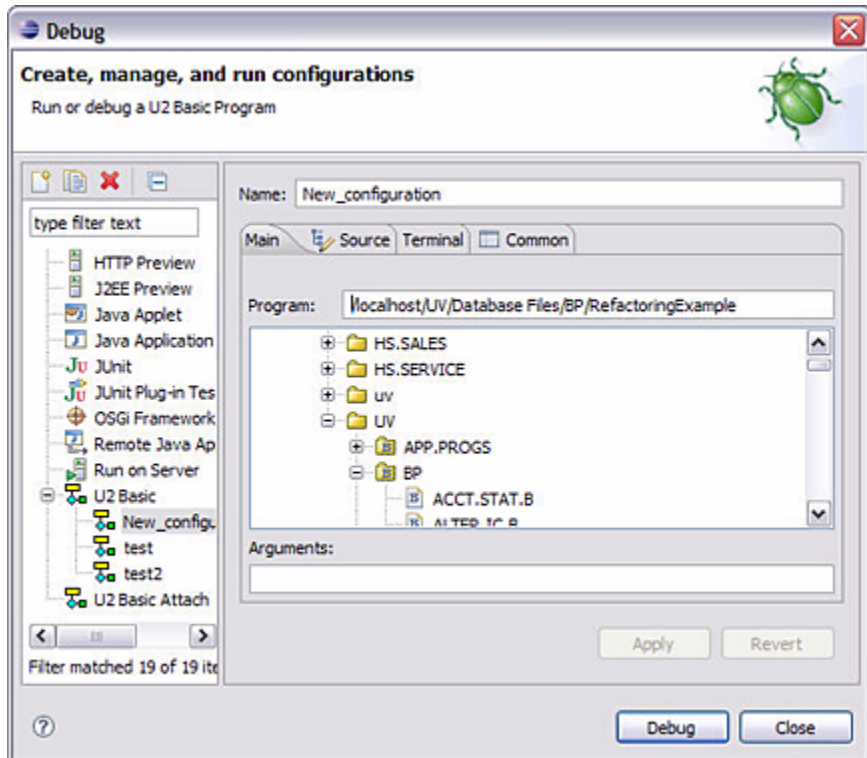


Debug icon

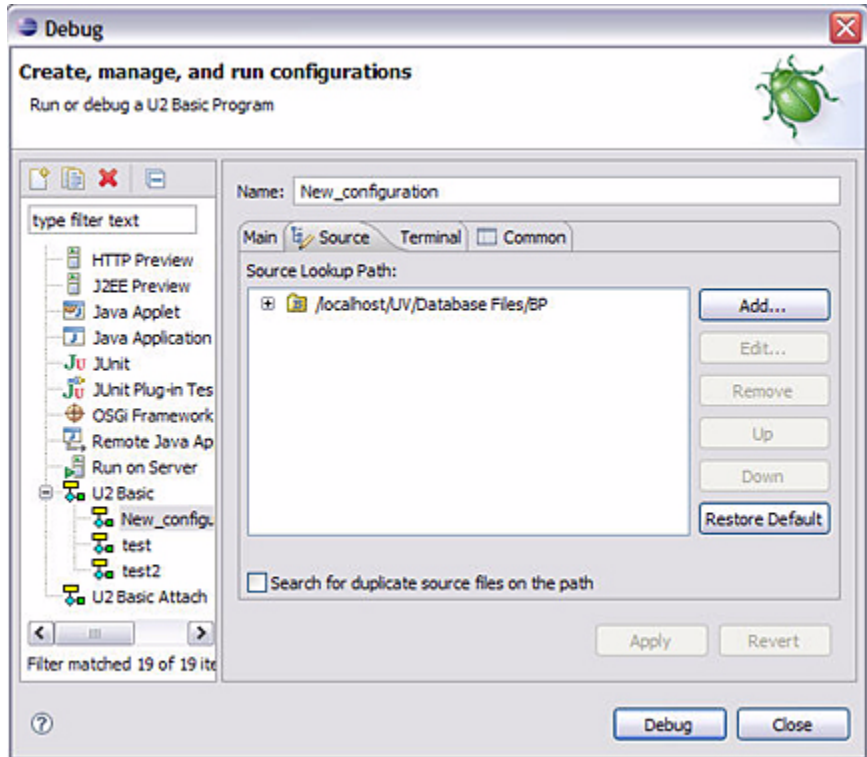
Click **Open Debug Dialog**. The **Debug** dialog box appears, as shown in the following example:



Click the **Main** tab. In the **Program** area of the **Debug** dialog box, expand the account where the UniVerse BASIC program that you want to debug resides, as shown in the following example:

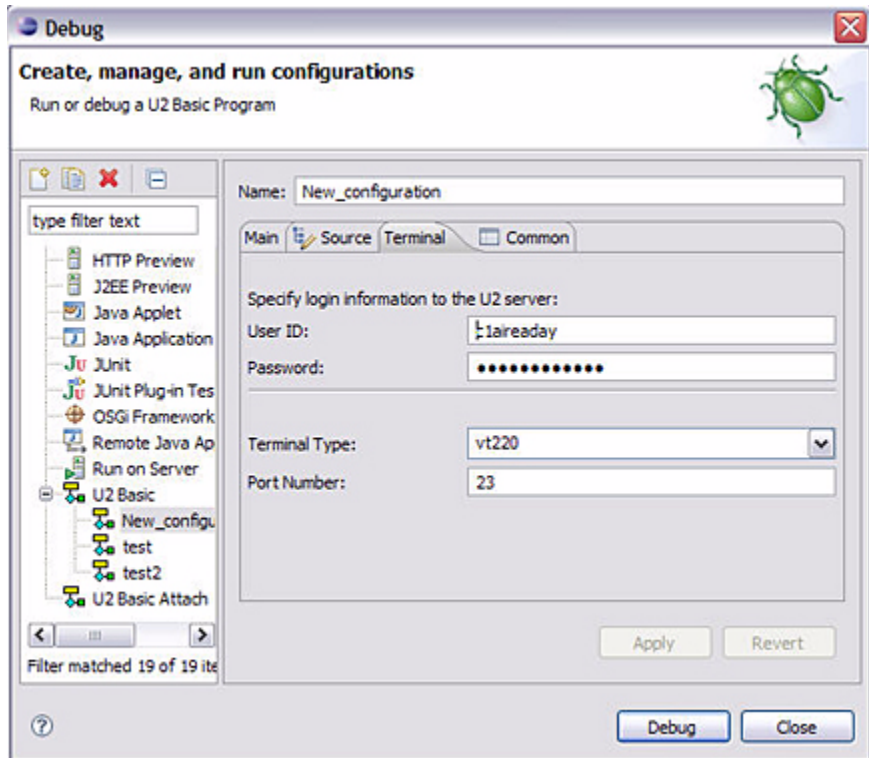


Click the **Source** tab. In this tab, the Basic Developer Toolkit displays directory files for the source locator to find the UniVerse BASIC source code. The following example illustrates the contents of the **Source** tab:



Click **Add** to include another path for the UniVerse BASIC source code.

Click the **Terminal** tab. A dialog box similar to the following example appears:



In the **User ID** box, enter the login ID on the server to which you want to connect. Enter the corresponding password in the **Password** box.

Select the terminal type you want to use in the **Terminal Type** box.

Enter the port number you want to use in the **Port Number** box.

Click **Debug**.

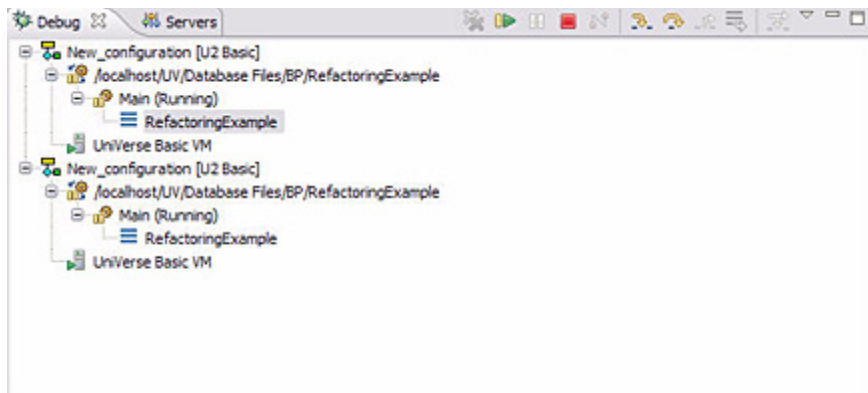
The Debug Perspective

When you debug a UniVerse BASIC program in the Basic Developer Toolkit, the debug perspective appears, consisting of the following views:

- [Stack View](#)
- Variable View
- Breakpoints View
- Outline View
- Console View
- U2 Console View
- U2 Debug Terminal View

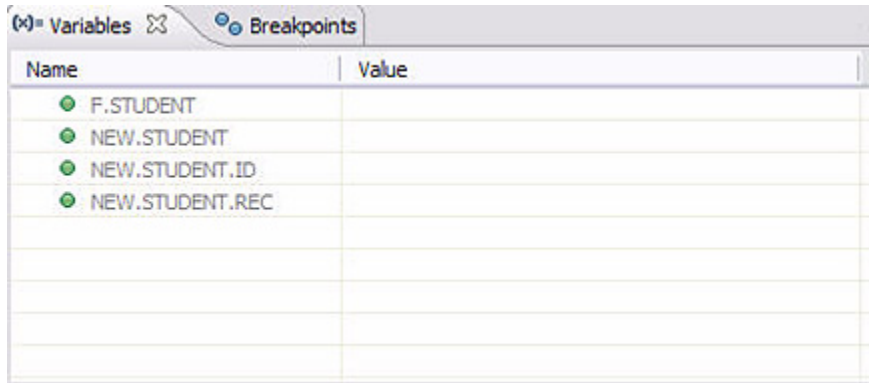
Stack View

The Stack View shows the current calling stack, as shown in the following example:



Variable View

The Variable View displays all the variables contained in the program and their values in the program that is currently running, as shown in the following example:



Name	Value
F.STUDENT	
NEW.STUDENT	
NEW.STUDENT.ID	
NEW.STUDENT.REC	

Due to performance issues, the Basic Developer Toolkit does not automatically monitor all variables. To enable a watch flag, right-click the variable you want to monitor, then click **Enable Watch Flag**. To disable a watch flag, right-click the variable you do not want to monitor and click **Disable Watch Flag**.

Right-click in the **Variables** view, then click **Show Monitored Variables Only** if you only want enabled variables to appear in the list.

Default Symbols and Words

A

This appendix describes the default symbols and words for the Basic Developer Toolkit code assist feature. If you do not specify a symbol or word in the XTOOLSUB.DATA file located in the *universe_home/include* directory, the code assist feature uses the default symbols and words described in this appendix.

Default Symbols

This section describes the default symbols used by the code assist feature of the Basic Developer Toolkit:

```
*symbols
*
{"@SYSTEM.RETURN.CODE", Vavvariable, 0}
{"@STDFIL", Vavvariable, 0}
{"@SELECTED", Vavvariable, 0}
{"@IM", Vprequate, 0, "Add: more ITM"}
{"@FM", Vprequate, 0}
{"@AM", Vprequate, 0}
{"@VM", Vprequate, 0}
{"@SM", Vprequate, 0}
{"@SVM", Vprequate, 0}
{"@TM", Vprequate, 0}
{"@CRTHIGH", Vprequate, 0}
{"@CRTWIDE", Vprequate, 0}
{"@LPTRHIGH", Vprequate, 0}
{"@LPTRWIDE", Vprequate, 0}
{"@WHO", Vrvariable, 0}
{"@LOGNAME", Vrvariable, 0}
{"@ACCOUNT", Vrvariable, 0}
{"@USERNO", Vrvariable, 0}
{"@USER.NO", Vrvariable, 0}
{"@COMMAND", Vrvariable, 0}
{"@SENTENCE", Vavvariable, 0}
{"@PARASENTENCE", Vrvariable, 0}
{"@SYS.BELL", Vrvariable, 0}
{"@LEVEL", Vrvariable, 0}
{"@TIME", Vavvariable, 0}
{"@DATE", Vavvariable, 0}
{"@DAY", Vavvariable, 0}
{"@MONTH", Vavvariable, 0}
{"@YEAR", Vavvariable, 0}
{"@FILENAME", Vavvariable, 0}
{"@FILE.NAME", Vavvariable, 0}
{"@RECORD", Vavvariable, 0}
{"@ID", Vavvariable, 0}
{"@USER.RETURN.CODE", Vavvariable, 0}
{"@SYSTEM.SET", Vavvariable, 0}
{"@TERM.TYPE", Vrvariable, 0}
{"@NB", Vavvariable, 0}
{"@ND", Vavvariable, 0}
{"@NI", Vavvariable, 0}
{"@RECCOUNT", Vavvariable, 0}
{"@NV", Vavvariable, 0}
{"@MV", Vavvariable, 0}
{"@NS", Vavvariable, 0}
{"@ANS", Vavvariable, 0}
{"@CONV", Vavvariable, 0}
{"@DICT", Vavvariable, 0}
{"@FORMAT", Vavvariable, 0}
{"@HEADER", Vavvariable, 0}
{"@OPTION", Vavvariable, 0}
{"@USER0", Vavvariable, 0}
```

```

{ "@USER1",                VavARIABLE,      0 }
{ "@USER2",                VavARIABLE,      0 }
{ "@USER3",                VavARIABLE,      0 }
{ "@USER4",                VavARIABLE,      0 }
{ "@RECUR0",               VavARIABLE,      0 }
{ "@RECUR1",               VavARIABLE,      0 }
{ "@RECUR2",               VavARIABLE,      0 }
{ "@RECUR3",               VavARIABLE,      0 }
{ "@RECUR4",               VavARIABLE,      0 }
{ "@TTY",                  VavARIABLE,      0 }
{ "@NULL",                 Vrvariable,      0 }
{ "@NULL.STR",             Vrvariable,      0 }
{ "@SQL.ERROR",            Vrvariable,      0 }
{ "@SQL.WARNING",          Vrvariable,      0 }
{ "@SQL.CODE",             Vrvariable,      0 }
{ "@SQL.STATE",            Vrvariable,      0 }
{ "@SCHEMA",               Vrvariable,      0 }
{ "@TRANSACTION.ID",       Vrvariable,      0 }
{ "@AUTHORIZATION",        Vrvariable,      0 }
{ "@ABORT.CODE",           Vrvariable,      0 }
{ "@COMMAND.STACK",        Vprequate,        0 }
{ "@DATA.PENDING",         Vprequate,        0 }
{ "@PATH",                 Vrvariable,      0 }
{ "@TRANSACTION",          Vrvariable,      0 }
{ "@ISOLATION",            Vrvariable,      0 }
{ "@TRANSACTION.LEVEL",    Vrvariable,      0 }
{ "@TRUE",                 Vprequate,        0 }
{ "@FALSE",                Vprequate,        0 }
{ "@HENV",                 Vrvariable,      0 }
{ "@HDBC",                 Vrvariable,      0 }
{ "@HSTMT",                Vrvariable,      0 }
{ "@SQLPROC.NAME",         Vrvariable,      0 }
{ "@SQLPROC.TX.LEVEL",     Vrvariable,      0 }
{ "@SQL.DATE",             Vrvariable,      0 }
{ "@SQL.TIME",             Vrvariable,      0 }
{ "@CASCADE",              Vrvariable,      0 }
{ "@YEAR4",                VavARIABLE,      0 }
{ "@DF.IOTYPE",            Vrvariable,      0 }

```

Default Words

This section describes the default words used by the code assist feature of the Basic Developer Toolkit:

```
*words
*
{"$DEFINE",          Lbos, 0, 0, 0, 0}
{"$ELSE",            Lcdir, 0, 0, 0, 0}
{"$ENDIF",           Lcdir, 0, 0, 0, 0}
{"$F",               0, 0, 0, 0, 0}
{"$FALSE",           0, 0, 0, 0, 0}
{"$IFDEF",           Lbig, 0, 0, 0, 0}
{"$IFNDEF",          Lbig, 0, 0, 0, 0}
{"$OPTIONS",         Lbos, 0, 0, 0, 0}
{"$T",               0, 0, 0, 0, 0}
{"$TRUE",            0, 0, 0, 0, 0}
{"$UNDEFINE",        Lbos, 0, 0, 0, 0}
{"ABORT",            Lbflag, 0, 0, 0, 0}
{"ABORTE",           Lbos, 0, 0, 0, 0}
{"ABORTM",           Lbos, 0, 0, 0, 0}
{"ABS",              Lfunct, 1, 1, 0, 0}
{"ABSS",             Lfunct, 1, 1, 0, 0}
{"acceptConnection", Lfunct, 6, 6, 0, LVA_4|LVA_5|LVA_6}
{"ACOS",             Lfunct, 1, 1, Ftrig, 0}
{"ACTIVATEKEY",      Lbos, 0, 0, 0, 0}
{"addAuthenticationRule", Lfunct, 4, 4, 0, 0}
{"addCertificate",    Lfunct, 5, 5, 0, 0}
{"addRequestParameter", Lfunct, 4, 4, 0, 0}
{"ADDS",             Lfunct, 2, 2, 0, 0}
{"ALL",              Lflag, 0, 0, 0, 0}
{"ALPHA",            Lfunct, 1, 1, 0, 0}
{"amInitialize",      Lfunct, 4, 4, 0, LVA_1|LVA_4}
{"amReceiveMsg",      Lfunct, 9, 10, 0, LVA_6|LVA_7|LVA_9}
{"amReceiveRequest", Lfunct, 9, 10, 0, LVA_5|LVA_6|LVA_9}
{"amSendMsg",         Lfunct, 6, 6, 0, LVA_6}
{"amSendRequest",     Lfunct, 7, 7, 0, LVA_7}
{"amSendResponse",    Lfunct, 7, 7, 0, LVA_7}
{"amTerminate",       Lfunct, 3, 3, 0, LVA_1|LVA_3}
{"analyzeCertificate", Lfunct, 3, 3, 0, 0}
{"AND",              0, 0, 0, 0, 0}
{"ANDS",             Lfunct, 2, 2, 0, 0}
{"ARG.",             Lflag, 0, 0, 0, 0}
{"ASCII",            Lfunct, 1, 1, 0, 0}
{"ASIN",             Lfunct, 1, 1, Ftrig, 0}
{"ASSIGN",           Lbos, 0, 0, 0, 0}
{"ASSIGNED",         Lfunct, 1, 1, 0, 0}
{"ATAN",             Lfunct, 1, 1, Ftrig, 0}
{"AUTHORIZATION",     Lbos, 2, 2, 0, 0}
{"AUTHORIZE",        Lfunct, 2, 2, 0, 0}
{"AUXMAP",           Lbos, 0, 0, 0, 0}
{"BCONVERT",         Lfunct, 2, 2, 0, 0}
{"BEFORE",           Lflag, 0, 0, 0, 0}
{"BEGIN",            Lbos, 0, 0, 0, 0}
{"BITAND",           Lfunct, 2, 2, 0, 0}
{"BITNOT",           Lfunct, 1, 2, Fnego, 0}
{"BITOR",            Lfunct, 2, 2, 0, 0}
```

```

{"BITRESET",          Lfunct, 2, 2, 0, 0}
{"BITSET",            Lfunct, 2, 2, 0, 0}
{"BITTEST",           Lfunct, 2, 2, 0, 0}
{"BITXOR",            Lfunct, 2, 2, 0, 0}
{"BREAK",             Lbos, 0, 0, 0, 0}
{"BSCAN",             Lbos, 0, 0, 0, 0}
{"BY",                Lflag, 0, 0, 0, 0}
{"BYTE",              Lfunct, 1, 1, 0, 0}
{"BYTELEN",           Lfunct, 1, 1, 0, 0}
{"BYTETYPE",          Lfunct, 1, 1, 0, 0}
{"BYTEVAL",           Lfunct, 1, 2, Fone, 0}
{"CALL",              Lbos, 0, 0, 0, 0}
{"CALLING",           Lflag, 0, 0, 0, 0}
{"CAPTURING",         Lflag, 0, 0, 0, 0}
{"CASE",              Lbflag, 0, 0, 0, 0}
{"CAT",               0, 0, 0, 0, 0}
{"CATS",              Lfunct, 2, 2, 0, 0}
{"CENTURY.PIVOT",     Lfunct, 1, 1, 0, 0}
{"CHAIN",             Lbos, 0, 0, 0, 0}
{"CHANGE",            Lfunct, 3, 5, 0, 0}
{"CHAR",              Lfunct, 1, 1, 0, 0}
{"CHARS",             Lfunct, 1, 1, 0, 0}
{"CHECKSUM",          Lfunct, 1, 1, 0, 0}
{"CLEAR",             Lbos, 0, 0, 0, 0}
{"CLEARCOMMON",       Lbos, 0, 0, 0, 0}
{"CLEARDATA",         Lbos, 0, 0, 0, 0}
{"CLEARDIAGNOSTICS",  Lfunct, 0, 0, 0, 0}
{"CLEARFILE",         Lbos, 0, 0, 0, 0}
{"CLEARINPUT",        Lbos, 0, 0, 0, 0}
{"CLEARPROMPTS",      Lbos, 0, 0, 0, 0}
{"CLEARSELECT",       Lbos, 0, 0, 0, 0}
{"CLOSE",             Lbflag, 0, 0, 0, 0}
{"CLOSESEQ",          Lbos, 0, 0, 0, 0}
{"closeSocket",       Lfunct, 1, 1, 0, LVA_1}
{"closeXMLData",      Lfunct, 1, 1, 0, 0}
{"COL1",              Lfunct, 0, 0, 0, 0}
{"COL2",              Lfunct, 0, 0, 0, 0}
{"COM",               Lbos, 0, 0, 0, 0}
{"COMMIT",            Lbflag, 0, 0, 0, 0}
{"COMMON",            Lbflag, 0, 0, 0, 0}
{"COMPARE",           Lfunct, 2, 3, Fnull, 0}
{"CONTINUE",          Lbos, 0, 0, 0, 0}
{"CONVERT",           Lbfunc, 3, 3, 0, 0}
{"COS",               Lfunct, 1, 1, Ftrig, 0}
{"COSH",              Lfunct, 1, 1, Ftrig, 0}
{"COUNT",            Lfunct, 2, 2, Fovlap, 0}
{"COUNTS",           Lfunct, 2, 2, 0, 0}
{"CREATE",            Lbos, 0, 0, 0, 0}
{"createCertificate",  Lfunct, 8, 8, 0, 0}
{"createCertRequest", Lfunct, 9, 9, 0, 0}
{"createRequest",     Lfunct, 3, 3, 0, LVA_3}
{"createSecureRequest", Lfunct, 4, 4, 0, LVA_3}
{"createSecurityContext", Lfunct, 2, 2, 0, LVA_1}
{"CRT",               Lbos, 0, 0, 0, 0}
{"DATA",              Lbos, 0, 0, 0, 0}
{"DATE",              Lfunct, 0, 0, 0, 0}
{"DCOUNT",            Lfunct, 2, 2, Fovlap, 0}
{"DEACTIVATEKEY",     Lbos, 0, 0, 0, 0}
{"DEBUG",             Lbos, 0, 0, 0, 0}
{"DECLARE",           Lbos, 0, 0, 0, 0}
{"DEFFUN",            Lbos, 0, 0, 0, 0}

```

```

{"DEL",
Lbos, 0, 0, 0, 0}
{"DELETE",
Lbfunc, 2, 4, 0, 0}
{"DELETEDLIST",
Lbos, 0, 0, 0, 0}
{"DELETEU",
Lbos, 0, 0, 0, 0}
{"DESCRINFO",
Lfuncnt, 2, 2, 0, 0}
{"DIAGNOSTICS",
Lfuncnt, 1, 2, 0, 0}
{"DIGEST",
Lfuncnt, 4, 4, 0, LVA_4}
{"DIM",
Lbos, 0, 0, 0, 0}
{"DIMENSION",
Lbos, 0, 0, 0, 0}
{"DISABLEDEC",
Lbos, 0, 0, 0, 0}
{"DISPLAY",
Lbos, 0, 0, 0, 0}
{"DIV",
Lfuncnt, 2, 2, 0, 0}
{"DIVS",
Lfuncnt, 2, 2, 0, 0}
{"DO",
Lflag, 0, 0, 0, 0}
{"DOWNCASE",
Lfuncnt, 1, 1, 0, 0}
{"DQUOTE",
Lfuncnt, 1, 1, 0, 0}
{"DTX",
Lfuncnt, 1, 2, 0, 0}
{"EBCDIC",
Lfuncnt, 1, 1, 0, 0}
{"ECHO",
Lbos, 0, 0, 0, 0}
{"ELSE",
Lelse, 0, 0, 0, 0}
{"ENABLEDEC",
Lbos, 0, 0, 0, 0}
{"ENCODE",
Lfuncnt, 6, 6, 0, LVA_5}
{"ENCRYPT",
Lfuncnt, 11, 11, 0, LVA_10}
{"END",
Lend, 0, 0, 0, 0}
{"ENTER",
Lbos, 0, 0, 0, 0}
{"EOF",
Lfuncnt, 0, 0, 0, 0}
{"EQ",
0, 0, 0, 0, 0}
{"EQS",
Lfuncnt, 2, 2, 0, 0}
{"EQU",
Lbos, 0, 0, 0, 0}
{"EQUATE",
Lbos, 0, 0, 0, 0}
{"EREPLACE",
Lfuncnt, 3, 5, 0, 0}
{"ERRMSG",
Lbos, 0, 0, 0, 0}
{"ERROR",
Lerror, 0, 0, 0, 0}
{"EXCHANGE",
Lfuncnt, 0, 0, 0, 0}
{"EXEC",
Lbos, 0, 0, 0, 0}
{"EXECUTE",
Lbos, 0, 0, 0, 0}
{"EXIT",
Lbos, 0, 0, 0, 0}
{"EXP",
Lfuncnt, 1, 1, 0, 0}
{"EXTRACT",
Lfuncnt, 2, 4, 0, 0}
{"FADD",
Lfuncnt, 2, 2, 0, 0}
{"FDIV",
Lfuncnt, 2, 2, 0, 0}
{"FFIX",
Lfuncnt, 1, 1, 0, 0}
{"FFLT",
Lfuncnt, 1, 1, 0, 0}
{"FIELD",
Lfuncnt, 3, 4, Fone, 0}
{"FIELDS",
Lfuncnt, 3, 4, Fone, 0}
{"FIELDSTORE",
Lfuncnt, 5, 5, 0, 0}
{"FILEINFO",
Lfuncnt, 2, 2, 0, 0}
{"FILELOCK",
Lbos, 0, 0, 0, 0}
{"FILEUNLOCK",
Lbos, 0, 0, 0, 0}
{"FIND",
Lbos, 0, 0, 0, 0}
{"FINDSTR",
Lbos, 0, 0, 0, 0}
{"FIX",
Lfuncnt, 1, 3, 0, 0}
{"FLUSH",
Lbos, 0, 0, 0, 0}
{"FMT",
Lfuncnt, 2, 2, 0, 0}
{"FMTDP",
Lfuncnt, 2, 3, Fnull, 0}
{"FMTS",
Lfuncnt, 2, 2, 0, 0}
{"FMTSDP",
Lfuncnt, 2, 3, Fnull, 0}
{"FMUL",
Lfuncnt, 2, 2, 0, 0}
{"FOLD",
Lfuncnt, 2, 2, 0, 0}
{"FOLDDP",
Lfuncnt, 2, 3, 0, 0}
{"FOOTING",
Lbos, 0, 0, 0, 0}

```



```

{"FOR",                      Lbos, 0, 0, 0, 0}
{"FORMLIST",                Lbos, 0, 0, 0, 0}
{"FROM",                    Lflag, 0, 0, 0, 0}
{"FSUB",                    Lfunct, 2, 2, 0, 0}
{"FUNCTION",                Lbos, 0, 0, 0, 0}
{"GARBAGECOLLECT",         Lbos, 0, 0, 0, 0}
{"GCI",                     Lflag, 0, 0, 0, 0}
{"GE",                      0, 0, 0, 0, 0}
{"generateKey",             Lfunct, 8, 8, 0, LVA_1|LVA_2}
{"GES",                     Lfunct, 2, 2, 0, 0}
{"GET",                     Lget, 0, 0, 0, 0}
{"getCipherSuite",         Lfunct, 2, 2, 0, LVA_2}
{"GETCONVERTTERMEURO",     Lfunct, 0, 0, 0, 0}
{"GETDIAGNOSTICS",         Lfunct, 0, 0, 0, 0}
{"getHTTPDefault",         Lfunct, 2, 2, 0, LVA_2}
{"GETLIST",                Lbos, 0, 0, 0, 0}
{"GETLOCALE",              Lfunct, 1, 1, 0, 0}
{"GETREM",                  Lfunct, 1, 1, 0, 0}
{"getResponseHeader",      Lfunct, 3, 3, 0, LVA_3}
{"getSocketErrorMessage",  Lfunct, 2, 2, 0, LVA_2}
{"getSocketInformation",   Lfunct, 3, 3, 0, LVA_3}
{"getSocketMap",           Lfunct, 2, 2, 0, LVA_2}
{"getSocketOptions",       Lfunct, 2, 2, 0, LVA_2}
{"GETSQLNULL",             Lfunct, 0, 0, 0, 0}
{"GETSYSTEMEURO",          Lfunct, 0, 0, 0, 0}
{"GETTERMEURO",            Lfunct, 0, 0, 0, 0}
{"GETX",                   Lbos, 0, 0, 0, 0}
{"GO",                      Lbflag, 0, 0, 0, 0}
{"GOSUB",                   Lbflag, 0, 0, 0, 0}
{"GOTO",                    Lbflag, 0, 0, 0, 0}
{"GROUP",                   Lfunct, 3, 4, Fone, 0}
{"GROUPSTORE",             Lbos, 0, 0, 0, 0}
{"GT",                      0, 0, 0, 0, 0}
{"GTS",                     Lfunct, 2, 2, 0, 0}
{"HEADING",                 Lbos, 0, 0, 0, 0}
{"HEADINGE",                Lbos, 0, 0, 0, 0}
{"HEADINGN",                Lbos, 0, 0, 0, 0}
{"HUSH",                    Lbos, 0, 0, 0, 0}
{"ICHECK",                  Lfunct, 2, 4, Fnego, 0}
{"ICONV",                   Lfunct, 2, 2, 0, 0}
{"ICONVS",                  Lfunct, 2, 2, 0, 0}
{"IF",                      0, 0, 0, 0, 0}
{"IFS",                     Lfunct, 3, 3, 0, 0}
{"ILPROMPT",                Lfunct, 1, 1, 0, 0}
{"IN",                      Lflag, 0, 0, 0, 0}
{"INDEX",                   Lfunct, 3, 3, Fovlap, 0}
{"INDEXS",                  Lfunct, 3, 3, 0, 0}
{"INDICES",                 Lfunct, 1, 2, Fnull, 0}
{"initSecureServerSocket", Lfunct, 5, 5, 0, LVA_4}
{"initServerSocket",        Lfunct, 4, 4, 0, LVA_4}
{"INMAT",                   Lfunct, 0, 0, 0, 0}
{"INPUT",                   Lbflag, 0, 0, 0, 0}
{"INPUTCLEAR",              Lbos, 0, 0, 0, 0}
{"INPUTDISP",               Lbos, 0, 0, 0, 0}
{"INPUTDP",                 Lbflag, 0, 0, 0, 0}
{"INPUTERR",                Lbos, 0, 0, 0, 0}
{"INPUTIF",                 Lbos, 0, 0, 0, 0}
{"INPUTNULL",               Lbos, 0, 0, 0, 0}
{"INPUTTRAP",               Lbos, 0, 0, 0, 0}
{"INS",                     Lbos, 0, 0, 0, 0}
{"INSERT",                  Lfunct, 3, 5, Fsemi, 0}

```

```

{"INT", Lfunc, 1, 1, 0, 0}
{"ISNULL", Lfunc, 1, 1, 0, 0}
{"ISNULLS", Lfunc, 1, 1, 0, 0}
{"ISOLATION", Lflag, 0, 0, 0, 0}
{"ITYPE", Lfunc, 1, 1, 0, 0}
{"KEY", Lflag, 0, 0, 0, 0}
{"KEYEDIT", Lbos, 0, 0, 0, 0}
{"KEYEXIT", Lbos, 0, 0, 0, 0}
{"KEYIN", Lfunc, 0, 0, 0, 0}
{"KEYTRAP", Lbos, 0, 0, 0, 0}
{"LE", 0, 0, 0, 0, 0}
{"LEFT", Lfunc, 2, 2, 0, 0}
{"LEN", Lfunc, 1, 1, 0, 0}
{"LENDP", Lfunc, 1, 2, Fnull, 0}
{"LENS", Lfunc, 1, 1, 0, 0}
{"LENSDP", Lfunc, 1, 2, Fnull, 0}
{"LES", Lfunc, 2, 2, 0, 0}
{"LET", Lbos, 0, 0, 0, 0}
{"LEVEL", Lflag, 0, 0, 0, 0}
{"LIT", Lflag, 0, 0, 0, 0}
{"LITERALLY", Lflag, 0, 0, 0, 0}
{"LN", Lfunc, 1, 1, 0, 0}
{"loadSecurityContext", Lfunc, 3, 3, 0, LVA_1}
{"LOCALEINFO", Lfunc, 1, 1, 0, 0}
{"LOCATE", Lloc, 0, 0, 0, 0}
{"LOCK", Lbos, 0, 0, 0, 0}
{"LOCKED", Llock, 0, 0, 0, 0}
{"LOOP", Lbos, 0, 0, 0, 0}
{"LOWER", Lfunc, 1, 1, 0, 0}
{"LPTR", Lflag, 0, 0, 0, 0}
{"LT", 0, 0, 0, 0, 0}
{"LTS", Lfunc, 2, 2, 0, 0}
{"MAT", Lbflag, 0, 0, 0, 0}
{"MATBUILD", Lbos, 0, 0, 0, 0}
{"MATCH", 0, 0, 0, 0, 0}
{"MATCHES", 0, 0, 0, 0, 0}
{"MATCHFIELD", Lfunc, 3, 3, 0, 0}
{"MATPARSE", Lbos, 0, 0, 0, 0}
{"MATREAD", Lbos, 0, 0, 0, 0}
{"MATREADL", Lbos, 0, 0, 0, 0}
{"MATREADU", Lbos, 0, 0, 0, 0}
{"MATWRITE", Lbos, 0, 0, 0, 0}
{"MATWRITEU", Lbos, 0, 0, 0, 0}
{"MAXIMUM", Lfunc, 1, 1, 0, 0}
{"MESSAGE", Lbos, 0, 0, 0, 0}
{"MINIMUM", Lfunc, 1, 1, 0, 0}
{"MOD", Lfunc, 2, 2, 0, 0}
{"MODS", Lfunc, 2, 2, 0, 0}
{"MTU", Lflag, 0, 0, 0, 0}
{"MULS", Lfunc, 2, 2, 0, 0}
{"NAP", Lbos, 0, 0, 0, 0}
{"NE", 0, 0, 0, 0, 0}
{"NEG", Lfunc, 1, 1, 0, 0}
{"NEGS", Lfunc, 1, 1, 0, 0}
{"NES", Lfunc, 2, 2, 0, 0}
{"NEXT", Lnext, 0, 0, 0, 0}
{"NO.ISOLATION", Lflag, 0, 0, 0, 0}
{"NOBUF", Lbos, 0, 0, 0, 0}
{"NOT", Lfunc, 1, 1, 0, 0}
{"NOTS", Lfunc, 1, 1, 0, 0}
{"NULL", Lbos, 0, 0, 0, 0}

```

```

{"NUM", Lfunct, 1, 1, 0, 0}
{"NUMS", Lfunct, 1, 1, 0, 0}
{"OCONV", Lfunct, 2, 2, 0, 0}
{"OCONVS", Lfunct, 2, 2, 0, 0}
{"OFF", Lflag, 0, 0, 0, 0}
{"ON", Lbflag, 0, 0, 0, 0}
{"OPEN", Lbos, 0, 0, 0, 0}
{"OPENCHECK", Lbos, 0, 0, 0, 0}
{"OPENDEV", Lbos, 0, 0, 0, 0}
{"OPENPATH", Lbos, 0, 0, 0, 0}
{"openSecureSocket", Lfunct, 6, 6, 0, LVA_5}
{"OPENSEQ", Lbos, 0, 0, 0, 0}
{"openSocket", Lfunct, 5, 5, 0, LVA_5}
{"OpenXMLData", Lfunct, 3, 3, 0, LVA_3}
{"OR", 0, 0, 0, 0, 0}
{"ORS", Lfunct, 2, 2, 0, 0}
{"OUT", Lflag, 0, 0, 0, 0}
{"PAGE", Lbos, 0, 0, 0, 0}
{"PASSLIST", Lflag, 0, 0, 0, 0}
{"PCDRIVER", Lbos, 0, 0, 0, 0}
{"PERFORM", Lexec, 0, 0, 0, 0}
{"PRECISION", Lbos, 0, 0, 0, 0}
{"PrepareXML", Lfunct, 2, 2, 0, LVA_2}
{"PRINT", Lbos, 0, 0, 0, 0}
{"PRINTER", Lbos, 0, 0, 0, 0}
{"PRINTERIO", Lbos, 0, 0, 0, 0}
{"PRINTERR", Lprntr, 0, 0, 0, 0}
{"PROCREAD", Lbos, 0, 0, 0, 0}
{"PROCWRITE", Lbos, 0, 0, 0, 0}
{"PROG", Lbos, 0, 0, 0, 0}
{"PROGRAM", Lbos, 0, 0, 0, 0}
{"PROMPT", Lbos, 0, 0, 0, 0}
{"protocolLogging", Lfunct, 3, 3, 0, 0}
{"PWR", Lfunct, 2, 2, 0, 0}
{"QUOTE", Lfunct, 1, 1, 0, 0}
{"RAISE", Lfunct, 1, 1, 0, 0}
{"RANDOMIZE", Lbos, 0, 0, 0, 0}
{"READ", Lbos, 0, 0, 0, 0}
{"READ.COMMITTED", Lflag, 0, 0, 0, 0}
{"READ.UNCOMMITTED", Lflag, 0, 0, 0, 0}
{"READBLK", Lbos, 0, 0, 0, 0}
{"READL", Lbos, 0, 0, 0, 0}
{"READLIST", Lbos, 0, 0, 0, 0}
{"READNEXT", Lbos, 0, 0, 0, 0}
{"READSEQ", Lbos, 0, 0, 0, 0}
{"readSocket", Lfunct, 6, 6, 0, LVA_2|LVA_6}
{"READT", Lbos, 0, 0, 0, 0}
{"READU", Lbos, 0, 0, 0, 0}
{"READV", Lbos, 0, 0, 0, 0}
{"READVL", Lbos, 0, 0, 0, 0}
{"READVU", Lbos, 0, 0, 0, 0}
{"ReadXMLData", Lfunct, 2, 2, 0, LVA_2}
{"REAL", Lfunct, 1, 1, 0, 0}
{"RECIO", Lfunct, 2, 4, 0, 0}
{"RECORDLOCKED", Lfunct, 2, 2, 0, 0}
{"RECORDLOCKL", Lbos, 0, 0, 0, 0}
{"RECORDLOCKU", Lbos, 0, 0, 0, 0}
{"RELEASE", Lbos, 0, 0, 0, 0}
{"ReleaseXML", Lfunct, 1, 1, 0, 0}
{"REM", Lfunct, 2, 2, 0, 0}
{"REMOVE", Lbfunc, 2, 2, 0, 0}

```

```

{"REPEAT",                Lloop, 0, 0, 0, 0}
{"REPEATABLE.READ",      Lflag, 0, 0, 0, 0}
{"REPLACE",              Lfunct, 3, 5, Fsemi, 0}
{"RESET",               Lflag, 0, 0, 0, 0}
{"RETURN",              Lsub, 0, 0, 0, 0}
{"RETURNING",           Lflag, 0, 0, 0, 0}
{"REUSE",               Lfunct, 1, 1, 0, 0}
{"REVREMOVE",           Lbos, 0, 0, 0, 0}
{"REWIND",              Lbos, 0, 0, 0, 0}
{"RIGHT",               Lfunct, 2, 2, 0, 0}
{"RND",                Lfunct, 1, 1, 0, 0}
{"ROLLBACK",            Lbos, 0, 0, 0, 0}
{"RPC.CALL",             Lfunct, 0, 0, 0, 0}
{"RPC.CONNECT",          Lfunct, 2, 2, 0, 0}
{"RPC.DISCONNECT",       Lfunct, 1, 1, 0, 0}
{"RQM",                 Lbos, 0, 0, 0, 0}
{"RTNLIST",             Lflag, 0, 0, 0, 0}
{"SADD",                Lfunct, 2, 2, 0, 0}
{"saveSecurityContext",  Lfunct, 3, 3, 0, 0}
{"SCMP",                Lfunct, 2, 2, 0, 0}
{"SDIV",                Lfunct, 2, 3, Fnego, 0}
{"SEEK",                Lseek, 0, 0, 0, 0}
{"SELECT",              Lbflag, 0, 0, 0, 0}
{"SELECTE",             Lbos, 0, 0, 0, 0}
{"SELECTINDEX",         Lbos, 0, 0, 0, 0}
{"SELECTINFO",          Lfunct, 2, 2, 0, 0}
{"SELECTN",             Lbos, 0, 0, 0, 0}
{"SELECTV",             Lbos, 0, 0, 0, 0}
{"SEND",                Lbos, 0, 0, 0, 0}
{"SENTENCE",            Lfunct, 0, 0, 0, 0}
{"SEQ",                 Lfunct, 1, 1, Fseq, 0}
{"SEQS",                Lfunct, 1, 1, 0, 0}
{"SEQSUM",              Lfunct, 1, 1, 0, 0}
{"SERIALIZABLE",        Lflag, 0, 0, 0, 0}
{"SET",                 Lbos, 0, 0, 0, 0}
{"setAuthenticationDepth", Lfunct, 3, 3, 0, 0}
{"setCipherSuite",       Lfunct, 2, 2, 0, 0}
{"setClientAuthentication", Lfunct, 2, 2, 0, 0}
{"SETCONVERTTERMEURO",  Lfunct, 1, 1, 0, 0}
{"SETDIAGNOSTICS",      Lfunct, 1, 3, Fnull, 0}
{"setHTTPDefault",       Lfunct, 2, 2, 0, 0}
{"SETLOCALE",           Lfunct, 2, 2, 0, 0}
{"setPrivateKey",        Lfunct, 6, 6, 0, 0}
{"setRandomSeed",        Lfunct, 4, 4, 0, 0}
{"SETREM",              Lbos, 0, 0, 0, 0}
{"setRequestHeader",     Lfunct, 3, 3, 0, 0}
{"setSocketMap",         Lfunct, 1, 1, 0, 0}
{"setSocketOptions",     Lfunct, 2, 2, 0, 0}
{"SETSYSTEMEURO",       Lfunct, 1, 1, 0, 0}
{"SETTERMEURO",         Lfunct, 1, 1, 0, 0}
{"SETTING",             Lflag, 0, 0, 0, 0}
{"showSecurityContext",  Lfunct, 2, 2, 0, 0}
{"SIGNATURE",           Lfunct, 10, 10, 0, LVA_9}
{"SIN",                 Lfunct, 1, 1, Ftrig, 0}
{"SINH",                Lfunct, 1, 1, Ftrig, 0}
{"SLEEP",               Lbos, 0, 0, 0, 0}
{"SMUL",                Lfunct, 2, 2, 0, 0}
{"SOAPCreateRequest",    Lfunct, 3, 3, 0, LVA_3}
{"SOAPCreateSecureRequest", Lfunct, 4, 4, 0, LVA_4}
{"SOAPGetDefault",       Lfunct, 2, 2, 0, LVA_2}
{"SOAPGetFault",         Lfunct, 2, 2, 0, LVA_2}

```

```

{"SOAPGetResponseHeader", Lfunct, 3, 3, 0, LVA_3}
{"SOAPRequestWrite",      Lfunct, 3, 3, 0, LVA_2}
{"SOAPSetDefault",       Lfunct, 2, 2, 0, 0}
{"SOAPSetParameters",    Lfunct, 4, 4, 0, 0}
{"SOAPSetRequestBody",    Lfunct, 2, 2, 0, 0}
{"SOAPSetRequestContent", Lfunct, 3, 3, 0, 0}
{"SOAPSetRequestHeader", Lfunct, 2, 2, 0, 0}
{"SOAPSubmitRequest",     Lfunct, 5, 5, 0, LVA_3|LVA_4|LVA_5}
{"SOUNDEX",               Lfunct, 1, 1, 0, 0}
{"SPACE",                 Lfunct, 1, 1, 0, 0}
{"SPACES",                Lfunct, 1, 1, 0, 0}
{"SPLICE",                Lfunct, 3, 3, 0, 0}
{"SQLALLOCCONNECT",      Lfunct, 2, 2, 0, LVA_2}
{"SQLALLOCENV",          Lfunct, 1, 1, 0, LVA_1}
{"SQLALLOCSTMT",         Lfunct, 2, 2, 0, LVA_2}
{"SQLBINDCOL",           Lfunct, 4, 4, 0, LVA_4}
{"SQLBINDPARAMETER",     Lfunct, 7, 8, Fone, LVA_7}
{"SQLCANCEL",            Lfunct, 1, 1, 0, 0}
{"SQLCMP",               Lfunct, 3, 3, Fnull, 0}
{"SQLCOLATTRIBUTES",     Lfunct, 5, 5, 0, LVA_4|LVA_5}
{"SQLCOLUMNS",          Lfunct, 5, 5, 0, 0}
{"SQLCONNECT",           Lfunct, 4, 4, 0, 0}
{"SQLDESCRIBECOL",       Lfunct, 7, 7, 0,
LVA_3|LVA_4|LVA_5|LVA_6|LVA_7}
{"SQLDISCONNECT",        Lfunct, 1, 1, 0, 0}
{"SQLERROR",             Lfunct, 6, 6, 0, LVA_4|LVA_5|LVA_6}
{"SQLEXECDIRECT",        Lfunct, 2, 2, 0, 0}
{"SQLEXECUTE",           Lfunct, 1, 1, 0, 0}
{"SQLFETCH",             Lfunct, 1, 1, 0, 0}
{"SQLFREECONNECT",       Lfunct, 1, 1, 0, 0}
{"SQLFREEENV",           Lfunct, 1, 1, 0, 0}
{"SQLFREESTMT",          Lfunct, 2, 2, 0, 0}
{"SQLGETCURSORNAME",     Lfunct, 2, 2, 0, LVA_2}
{"SQLGETINFO",           Lfunct, 3, 3, 0, LVA_3}
{"SQLGETTYPEINFO",       Lfunct, 2, 2, 0, 0}
{"SQLNUMPARAMS",         Lfunct, 2, 2, 0, LVA_2}
{"SQLNUMRESULTCOLS",     Lfunct, 2, 2, 0, LVA_2}
{"SQLPARAMOPTIONS",      Lfunct, 3, 3, 0, LVA_3}
{"SQLPREPARE",           Lfunct, 2, 2, 0, 0}
{"SQLROWCOUNT",         Lfunct, 2, 2, 0, LVA_2}
{"SQLSETCONNECTOPTION",  Lfunct, 3, 3, 0, 0}
{"SQLSETCURSORNAME",     Lfunct, 2, 2, 0, 0}
{"SQLSETPARAM",          Lfunct, 7, 8, Fone, LVA_7}
{"SQLSPECIALCOLUMNS",  Lfunct, 7, 7, 0, 0}
{"SQLSTATISTICS",        Lfunct, 6, 6, 0, 0}
{"SQLTABLES",            Lfunct, 5, 5, 0, 0}
{"SQLTRANSACT",          Lfunct, 3, 3, 0, 0}
{"SQRT",                 Lfunct, 1, 1, 0, 0}
{"SQUOTE",               Lfunct, 1, 1, 0, 0}
{"SSELECT",              Lbos, 0, 0, 0, 0}
{"SSELECTN",             Lbos, 0, 0, 0, 0}
{"SSELECTV",             Lbos, 0, 0, 0, 0}
{"SSUB",                 Lfunct, 2, 2, 0, 0}
{"START",                Lflag, 0, 0, 0, 0}
{"STATUS",               Lstat, 0, 0, 0, 0}
{"STEP",                 Lflag, 0, 0, 0, 0}
{"STOP",                 Lbos, 0, 0, 0, 0}
{"STOPE",                Lbos, 0, 0, 0, 0}
{"STOPM",                Lbos, 0, 0, 0, 0}
{"STORAGE",              Lbos, 0, 0, 0, 0}
{"STR",                  Lfunct, 2, 2, 0, 0}

```

```

{"STRS",                Lfuncnt, 2, 2, 0, 0}
{"SUB",                 Lsub, 0, 0, 0, 0}
{"submitRequest",       Lfuncnt, 6, 6, 0, LVA_4|LVA_5|LVA_6}
{"SUBR",                Lfuncnt, 0, 0, 0, 0}
{"SUBROUTINE",          Lsub, 0, 0, 0, 0}
{"SUBS",                Lfuncnt, 2, 2, 0, 0}
{"SUBSTRINGS",          Lfuncnt, 3, 3, 0, 0}
{"SUM",                 Lfuncnt, 1, 1, 0, 0}
{"SUMMATION",           Lfuncnt, 1, 1, 0, 0}
{"SWAP",                Lbos, 2, 2, 0, 0}
{"SYSTEM",              Lfuncnt, 1, 2, 0, 0}
{"TABSTOP",             Lbos, 0, 0, 0, 0}
{"TAN",                 Lfuncnt, 1, 1, Ftrig, 0}
{"TANH",                Lfuncnt, 1, 1, Ftrig, 0}
{"TERMINFO",            Lfuncnt, 1, 1, 0, 0}
{"TESTDIAGNOSTICS",     Lfuncnt, 0, 0, 0, 0}
{"THEN",                Lthen, 0, 0, 0, 0}
{"TIME",                Lfuncnt, 0, 0, 0, 0}
{"TIMEDATE",            Lfuncnt, 0, 0, 0, 0}
{"TIMEOUT",             Lbos, 0, 0, 0, 0}
{"TO",                  Lflag, 0, 0, 0, 0}
{"TPARM",               Lfuncnt, 1, 9, 0, 0}
{"TPRINT",              Lbos, 0, 0, 0, 0}
{"TRANS",               Lfuncnt, 4, 4, 0, 0}
{"TRANSACTION",         Lbflag, 0, 0, 0, 0}
{"TRIM",                Lfuncnt, 1, 3, 0, 0}
{"TRIMB",               Lfuncnt, 1, 1, 0, 0}
{"TRIMBS",              Lfuncnt, 1, 1, 0, 0}
{"TRIMF",               Lfuncnt, 1, 1, 0, 0}
{"TRIMFS",              Lfuncnt, 1, 1, 0, 0}
{"TRIMS",               Lfuncnt, 1, 1, 0, 0}
{"TTYCTL",              Lbos, 0, 0, 0, 0}
{"TTYGET",              Lbos, 0, 0, 0, 0}
{"TTYSET",              Lbos, 0, 0, 0, 0}
{"UNASSIGNED",          Lfuncnt, 1, 1, 0, 0}
{"UNICHAR",             Lfuncnt, 1, 1, 0, 0}
{"UNICHARS",            Lfuncnt, 1, 1, 0, 0}
{"UNISEQ",              Lfuncnt, 1, 1, 0, 0}
{"UNISEQS",             Lfuncnt, 1, 1, 0, 0}
{"UNIT",                Lfuncnt, 0, 0, 0, 0}
{"UNLOCK",              Lbos, 0, 0, 0, 0}
{"UNTIL",               Lloop, 0, 0, 0, 0}
{"UPCASE",              Lfuncnt, 1, 1, 0, 0}
{"UPRINT",              Lbos, 0, 0, 0, 0}
{"USING",               Lflag, 0, 0, 0, 0}
{"WAITING",             Lflag, 0, 0, 0, 0}
{"WEOF",                Lbos, 0, 0, 0, 0}
{"WEOFSEQ",             Lbos, 0, 0, 0, 0}
{"WHILE",               Lloop, 0, 0, 0, 0}
{"WORDSIZE",            Lbos, 0, 0, 0, 0}
{"WORK",                Lflag, 0, 0, 0, 0}
{"WRITE",               Lbos, 0, 0, 0, 0}
{"WRITEBLK",            Lbos, 0, 0, 0, 0}
{"WRITELIST",           Lbos, 0, 0, 0, 0}
{"WRITESEQ",            Lbos, 0, 0, 0, 0}
{"WRITESEQF",           Lbos, 0, 0, 0, 0}
{"writeSocket",         Lfuncnt, 5, 5, 0, LVA_5}
{"WRITET",              Lbos, 0, 0, 0, 0}
{"WRITEU",              Lbos, 0, 0, 0, 0}
{"WRITEV",              Lbos, 0, 0, 0, 0}
{"WRITEVU",             Lbos, 0, 0, 0, 0}

```

```

{"XDOMAddChild",          Lfunct, 5, 5, 0, 0}
{"XDOMAppend",           Lfunct, 5, 5, 0, 0}
{"XDOMClone",            Lfunct, 3, 3, 0, LVA_2}
{"XDOMClose",            Lfunct, 1, 1, 0, 0}
{"XDOMCreateNode",       Lfunct, 5, 5, 0, LVA_5}
{"XDOMCreateRoot",       Lfunct, 1, 2, 0, LVA_1}
{"XDOMEvaluate",         Lfunct, 4, 4, 0, LVA_4}
{"XDOMGetAttribute",     Lfunct, 3, 3, 0, LVA_3}
{"XDOMGetNodeName",      Lfunct, 2, 2, 0, LVA_2}
{"XDOMGetNodeType",      Lfunct, 2, 2, 0, LVA_2}
{"XDOMGetNodeValue",     Lfunct, 2, 2, 0, LVA_2}
{"XDOMGetOwnerDocument", Lfunct, 2, 2, 0, LVA_2}
{"XDOMGetUserData",      Lfunct, 2, 2, 0, LVA_2}
{"XDOMImportNode",       Lfunct, 4, 4, 0, LVA_4}
{"XDOMInsert",           Lfunct, 5, 5, 0, 0}
{"XDOMLocate",           Lfunct, 4, 4, 0, LVA_4}
{"XDOMLocateNode",       Lfunct, 5, 5, 0, LVA_5}
{"XDOMOpen",             Lfunct, 3, 3, 0, LVA_3}
{"XDOMRemove",           Lfunct, 5, 5, 0, LVA_5}
{"XDOMReplace",          Lfunct, 5, 5, 0, 0}
{"XDOMSetNodeValue",     Lfunct, 2, 2, 0, 0}
{"XDOMSetUserData",      Lfunct, 2, 2, 0, 0}
{"XDOMTransform",        Lfunct, 4, 5, 0, LVA_4}
{"XDOMValidate",         Lfunct, 4, 5, 0, 0}
{"XDOMValidateDom",      Lfunct, 3, 4, 0, 0}
{"XDOMWrite",            Lfunct, 3, 4, 0, LVA_2}

{"XLATE",                Lfunct, 4, 4, 0, 0}
{"XMAPAppendRec",        Lfunct, 3, 3, 0, 0}
{"XMAPClose",            Lfunct, 1, 1, 0, 0}
{"XMAPCreate",           Lfunct, 3, 3, 0, LVA_3}
{"XMAPGetInfo",          Lfunct, 2, 2, 0, LVA_2}
{"XMAPOpen",             Lfunct, 5, 5, 0, LVA_5}
{"XMAPReadNext",         Lfunct, 3, 3, 0, LVA_3}
{"XMAPSetOption",        Lfunct, 3, 3, 0, 0}
{"XMAPToXMLDoc",         Lfunct, 3, 4, 0, LVA_2}

{"XMLERROR",             Lfunct, 1, 1, 0, LVA_1}
{"XMLExecute",           Lfunct, 4, 4, 0, LVA_3|LVA_4}
{"XMLGetError",          Lfunct, 2, 2, 0, LVA_1|LVA_2}

{"XMLGetOptions",        Lfunct, 1, 2, 0, LVA_1}
{"XMLGetOptionValue",    Lfunct, 2, 2, 0, LVA_1}
{"XMLSetOptions",        Lfunct, 1, 1, 0, 0, "Since UV

10.3"}

{"XTD",                  Lfunct, 1, 1, 0, 0}

```