

SYBASE®

Component Tutorials: Adaptive Server Enterprise

**Sybase® WorkSpace**

1.5

DOCUMENT ID: DC00508-01-0150-01

LAST REVISED: June 2006

Copyright © 2005-2006 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase, SYBASE (logo), ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Advantage Database Server, Afaria, Answers Anywhere, Applied Meta, Applied Metacomputing, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Translator, APT-Library, ASEP, Avaki, Avaki (Arrow Design), Avaki Data Grid, AvantGo, Backup Server, BayCam, Beyond Connected, Bit-Wise, BizTracker, Certified PowerBuilder Developer, Certified SYBASE Professional, Certified SYBASE Professional Logo, ClearConnect, Client-Library, Client Services, CodeBank, Column Design, ComponentPack, Connection Manager, Convoy/DM, Copernicus, CSP, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DataWindow .NET, DB-Library, dbQueue, Dejima, Dejima Direct, Developers Workbench, DirectConnect Anywhere, DirectConnect, Distribution Director, Dynamic Mobility Model, e-ADK, E-Anywhere, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, EII Plus, Electronic Case Management, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise Portal (logo), Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, eremote, Everything Works Better When Everything Works Together, EWA, ExtendedAssist, Extended Systems, ExtendedView, Financial Fusion, Financial Fusion (and design), Financial Fusion Server, Formula One, Fusion Powered e-Finance, Fusion Powered Financial Destinations, Fusion Powered STP, Gateway Manager, GeoPoint, GlobalFIX, iAnywhere, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InstaHelp, Intelligent Self-Care, InternetBuilder, iremote, irLite, iScript, Jaguar CTS, jConnect for JDBC, KnowledgeBase, Legion, Logical Memory Manager, M2M Anywhere, Mach Desktop, Mail Anywhere Studio, Mainframe Connect, Maintenance Express, Manage Anywhere Studio, MAP, M-Business Anywhere, M-Business Channel, M-Business Network, M-Business Suite, MDI Access Server, MDI Database Gateway, media.splash, Message Anywhere Server, MetaWorks, MethodSet, mFolio, Mirror Activator, ML Query, MobiCATS, MobileQ, MySupport, Net-Gateway, Net-Library, New Era of Networks, Next Generation Learning, Next Generation Learning Studio, O DEVICE, OASIS, OASIS logo, ObjectConnect, ObjectCycle, OmniConnect, OmniQ, OmniSQL Access Module, OmniSQL Toolkit, OneBridge, Open Biz, Open Business Interchange, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Partnerships that Work, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, Pharma Anywhere, PhysicalArchitect, Pocket PowerBuilder, PocketBuilder, Power++, Power Through Knowledge, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, Powering the New Economy, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Pylon, Pylon Anywhere, Pylon Application Server, Pylon Conduit, Pylon PIM Server, Pylon Pro, QAnywhere, Rapport, Relational Beans, RemoteWare, RepConnector, Report Workbench, Report-Execute, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Resource Manager, RFID Anywhere, RW-DisplayLib, RW-Library, SAFE, SAFE/PRO, Sales Anywhere, Search Anywhere, SDF, Search Anywhere, Secure SQL Server, Secure SQL Toolset, Security Guardian, ShareSpool, ShareLink, SKILS, smart.partners, smart.parts, smart.script, SOA Anywhere Trademark, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, Stage III Engineering, Startup.Com, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Development Framework, Sybase Financial Server, Sybase Gateways, Sybase IQ, Sybase Learning Connection, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase Synergy Program, Sybase Virtual Server Architecture, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SybFlex, SybMD, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, The Enterprise Client/Server Company, The Extensible Software Platform, The Future Is Wide Open, The Learning Connection, The Model For Client/Server Solutions, The Online Information Center, The Power of One, TotalFix, TradeForce, Transact-SQL, Translation Toolkit, Turning Imagination Into Reality, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Viafone, Viewer, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server, XcelleNet, XP Server, XTNDAccess and XTNDConnect are trademarks of Sybase, Inc. or its subsidiaries. 05/06

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

# Contents

<b>About This Book .....</b>	<b>v</b>	
<b>CHAPTER 1</b>	<b>Introduction to ASE Component Tutorials .....</b>	<b>1</b>
	Setup .....	1
	Creating a connection profile for an ASE server .....	2
	Installing the interpubs database .....	5
<b>CHAPTER 2</b>	<b>Stored Procedures Tutorial .....</b>	<b>9</b>
	Before you begin .....	9
	Working with stored procedures .....	10
	Lesson 1: View existing stored procedures .....	10
	Lesson 2: Create a stored procedure .....	14
	Lesson 3: Run a stored procedure .....	21
	Lesson 4: View #temp tables during debugging .....	23
<b>CHAPTER 3</b>	<b>Triggers Tutorials .....</b>	<b>33</b>
	Before you begin .....	33
	Viewing and creating triggers .....	33
	Lesson 1: View an existing trigger .....	34
	Lesson 2: Create a new trigger .....	36
	Running and debugging a trigger .....	40
	Lesson 1: Run a trigger .....	40
	Lesson 2: View the inserted special table within a trigger .....	42
	Lesson 3: Debug the trigger .....	45



# About This Book

## Audience

This guide is intended for users who want to learn how to use Sybase® WorkSpace integrated development tooling with Adaptive Server® Enterprise to create service-oriented Web applications.

## How to use this book

This guide contains these chapters:

- Chapter 1, “Introduction to ASE Component Tutorials” introduces the Sybase WorkSpace component tutorials for Adaptive Server Enterprise and describes the setup and installation tasks you must perform before you can run the tutorials.
- Chapter 2, “Stored Procedures Tutorial” demonstrates how to use Sybase WorkSpace to view, create, and run stored procedures for an Adaptive Server Enterprise database.
- Chapter 3, “Triggers Tutorials” demonstrates how to use Sybase WorkSpace to view, create, and debug triggers on an Adaptive Server Enterprise server.

## Related documents

For more information on Sybase WorkSpace and Adaptive Server Enterprise:

**Sybase WorkSpace online bookshelf** From the Sybase WorkSpace, main menu, select **Help|Help Contents** to view the Sybase WorkSpace and supporting documentation.

The tutorial and sample files and documentation are available for download from Sybase CodeXchange. From the Sybase WorkSpace main menu, select **Help|Tutorials** for more information.

**Adaptive Server Enterprise online bookshelf** See online product documentation at <http://sybooks.sybase.com/>.

**Sybase WorkSpace Getting Started CD** The Sybase WorkSpace Getting Started CD includes the following documents:

*Sybase WorkSpace 1.5 Installation Guide*

*Sybase Developer Edition Servers 1.5 Installation Guide*

*Sybase WorkSpace 1.5 Release Bulletin*

---

*Adaptive Server Enterprise 15.0 Installation Guide*

*Unwired Accelerator 7.0 Installation Guide*

To access the Sybase Product Manuals Web site, go to Product Manuals at <http://sybooks.sybase.com/>

**Other sources of information**

Use the Sybase Getting Started CD, the SyBooks™ CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

To access the Sybase Product Manuals Web site, go to Product Manuals at <http://sybooks.sybase.com/>.

**Sybase certifications on the Web**

Technical documentation at the Sybase Web site is updated frequently.

❖ **Finding the latest information on product certifications**

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Click Certification Report.
- 3 In the Certification Report filter select a product, platform, and timeframe and then click Go.

- 4 Click a Certification Report title to display the report.

❖ **Finding the latest information on component certifications**

- 1 Point your Web browser to Availability and Certification Reports at <http://certification.sybase.com/>.
- 2 Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.
- 3 Select Search to display the availability and certification report for the selection.

❖ **Creating a personalized view of the Sybase Web site (including support pages)**

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Click MySybase and create a MySybase profile.

**Sybase EBFs and software maintenance**

❖ **Finding the latest information on EBFs and software maintenance**

- 1 Point your Web browser to the Sybase Support Page at <http://www.sybase.com/support>.
- 2 Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
- 3 Select a product.
- 4 Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.

- 5 Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

---

## Conventions

The following formatting conventions are used in this manual:

Formatting example	To indicate
command names and method names	When used in descriptive text, this font indicates keywords such as: <ul style="list-style-type: none"><li>• Command names used in descriptive text</li><li>• C++ and Java method or class names used in descriptive text</li><li>• Java package names used in descriptive text</li></ul>
<i>myCounter</i> variable <i>server.log</i> <i>myfile.txt</i>	Italic font indicates: <ul style="list-style-type: none"><li>• Program variables</li><li>• Parts of input text that must be substituted</li><li>• Directory and file names</li></ul>
<i>sybase\bin</i>	A backward slash (“\”) indicates cross-platform directory information. A forward slash (“/”) applies to information specific only to UNIX. Directory names appearing in text display in lowercase unless the system is case sensitive.
<b>File Save</b>	Menu names and menu items are displayed in plain text. The pipe indicates how to navigate menu selections, such as from the File menu to the Save option.
parse put get Name Address	The vertical bar indicates: <ul style="list-style-type: none"><li>• Options available within code</li><li>• Delimiter within message examples</li></ul>
create table table created	Monospace font indicates: <ul style="list-style-type: none"><li>• Information that you enter on a command line or as program text.</li><li>• Example output fragments</li></ul>
Type the <b>Name</b> of the attribute. Click <b>Apply</b> .	GUI field or button name that is the recipient of a procedural action.
setup -is:tempdir <full path to alternate temp directory>	Information that must be supplied by the user is displayed between brackets.

## If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.



# Introduction to ASE Component Tutorials

Sybase WorkSpace fully integrates with Adaptive Server Enterprise to enable you to view and manage your data and build enterprise applications. You can use Sybase WorkSpace tooling to run and debug procedural objects including stored procedures, triggers, events, and user-defined functions.

The Sybase WorkSpace Adaptive Server Enterprise tutorials demonstrate how to view and create two key procedural objects: stored procedures and triggers.

For additional background information on the features or concepts for Adaptive Server Enterprise, Sybase WorkSpace, or the Eclipse integrated development environment (IDE), see “Related documents” on page v.

Before you can perform the Sybase WorkSpace component tutorials for Adaptive Server Enterprise, you must complete the prerequisite tasks, described in “Setup” below.

## Setup

Before you can perform the Sybase WorkSpace component tutorials for Adaptive Server Enterprise,

- Install Sybase WorkSpace 1.5 with the Database Development tooling. See the *Sybase WorkSpace Installation Guide* on the Sybase WorkSpace Getting Started CD or the Sybase Product Manuals Web site at <http://www.sybase.com/support/techdocs>.

If you have not purchased Sybase WorkSpace but would like to perform this tutorial, you can download an evaluation version at <http://eshop.sybase.com/eshop/>. Once you log in to eShop, in the left pane, click **Development & Integration** and then **WorkSpace**.

- Install an Adaptive Server Enterprise server and start it. You must have a Sybase-WorkSpace-supported version of Adaptive Server Enterprise installed and running (version 12.5.3 or later).

If you have not purchased Adaptive Server Enterprise but would like to perform this tutorial, you can download an evaluation version of Sybase WorkSpace, which includes Adaptive Server Enterprise, at <http://eshop.sybase.com/eshop/>. Once you log in to eShop, in the left pane, click **Development & Integration** and then **WorkSpace**.

- Have basic knowledge of the Eclipse IDE and the database perspectives in Sybase WorkSpace.
- Create a connection profile, which enables connection to servers, message transports, and databases. See “Creating a connection profile for an ASE server” on page 2.
- Install an Adaptive Server Enterprise pubs sample database. You can use *pubs2*, *pubs3*, or *interpubs*, which are similar to each other with minor differences. This tutorial uses the *interpubs* sample database. See “Installing the interpubs database” on page 5 for more information.

## Creating a connection profile for an ASE server

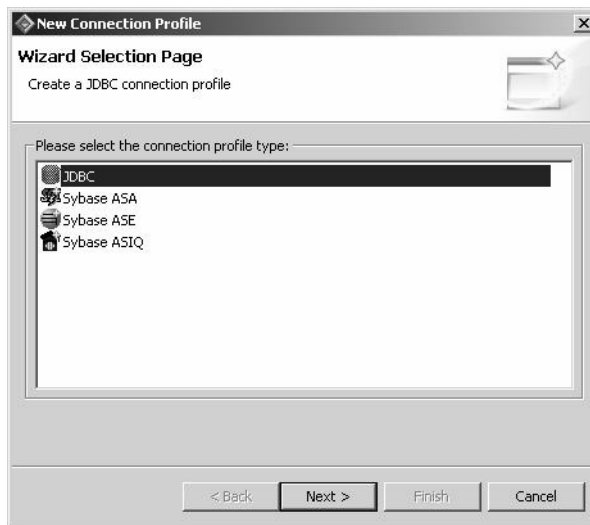
This section demonstrates how to create a connection profile for an Adaptive Server Enterprise server in Sybase WorkSpace. A connection profile enables connection to servers, message transports, and databases.

- 1 Launch Sybase WorkSpace.

You create the connection profile in the Database Development perspective.

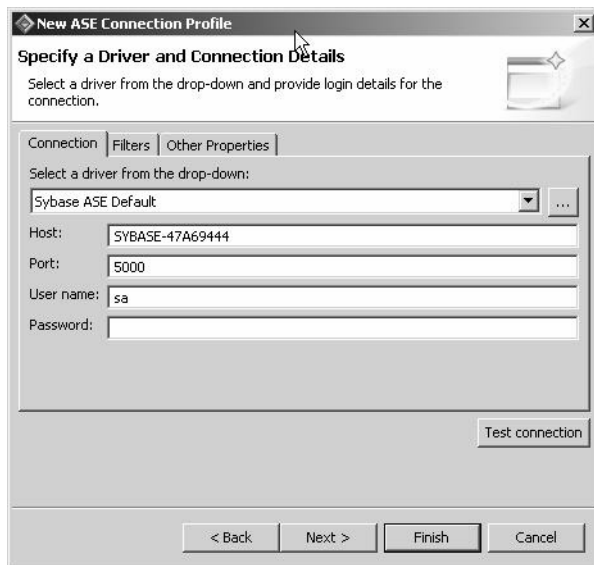
- 2 To open the **Database Development** perspective, select **Window|Open Perspective|Database Development** from the Sybase WorkSpace main menu bar.
- 3 In the **Database Explorer** view, right-click **Database Connection Profiles** and select **New Connection Profile**.

The **New Connection Profile** wizard displays.



- 4 Select **Sybase ASE** and click **Next**.
- 5 In the **Name** field, enter `ASE_Tutorials` and click **Next**.
- 6 In the **Connection** tab, in the Host and Port fields, enter the proper host (server name) and its associated port number.

If you do not know the host name or port number, check the sql.ini file (located in C:\Sybase\ini by default), which lists network servers and their associated port numbers, or see your system administrator.



- 7 Enter your database user name and password.

Your user account must have privileges to access the sample pubs database. If you do not know your user name and/or password, see your system administrator.

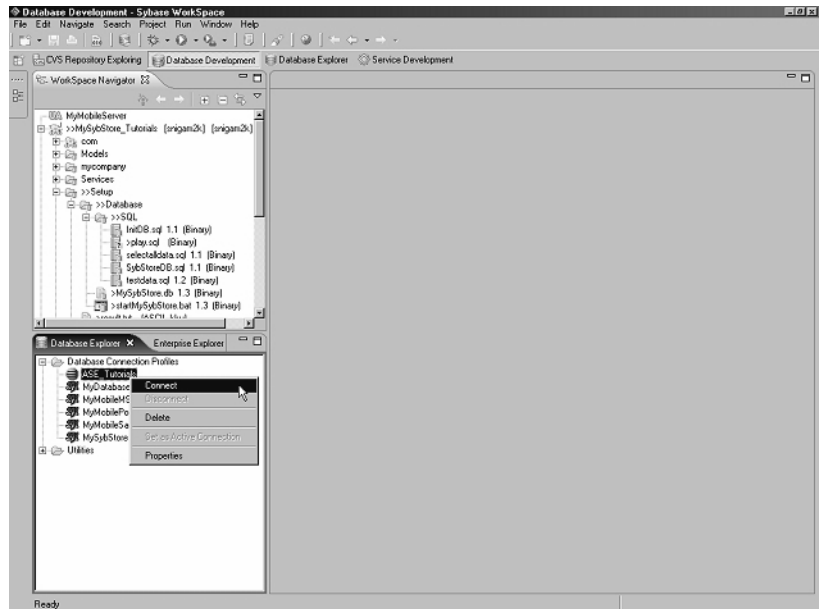
- 8 Click **Test connection** to verify you entered the proper server information.



Do not proceed until the ping succeeds. If the ping does not succeed, your user account might not have privileges to access the sample pubs database. See your system administrator for assistance.

- 9 Click **OK** to close the **Success** dialog box and then click **Finish**.
- 10 If you have not installed a samples database, continue to “Installing the interpubs database” on page 5.

If you already installed a samples database, in the **Database Explorer** view, right-click the **ASE\_Tutorials** connection profile you just created and select **Connect**.



You have successfully created a connection profile for an Adaptive Server Enterprise server and are connected to the server from Sybase WorkSpace.

## Installing the *interpubs* database

The Adaptive Server Enterprise tutorials use the *interpubs* sample database in their examples; however, you can also use the *pubs2* and *pubs3* sample databases.

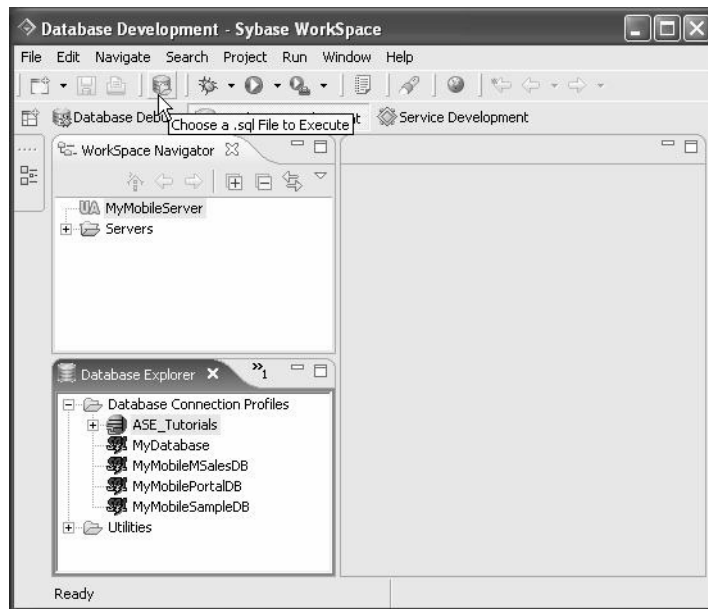
The following instructions show you how to install this database on the master device; however, your organization might prefer you to install the database on a different device due to resource constraints. For information on how to install the *interpubs* sample database on another device, or to install the *pubs2* or *pubs3* database, see the *Adaptive Server Enterprise Installation Guide* or your system administrator.

You can install the samples database using Sybase WorkSpace or a command window.

❖ **Installing the interpubs database using Sybase WorkSpace**

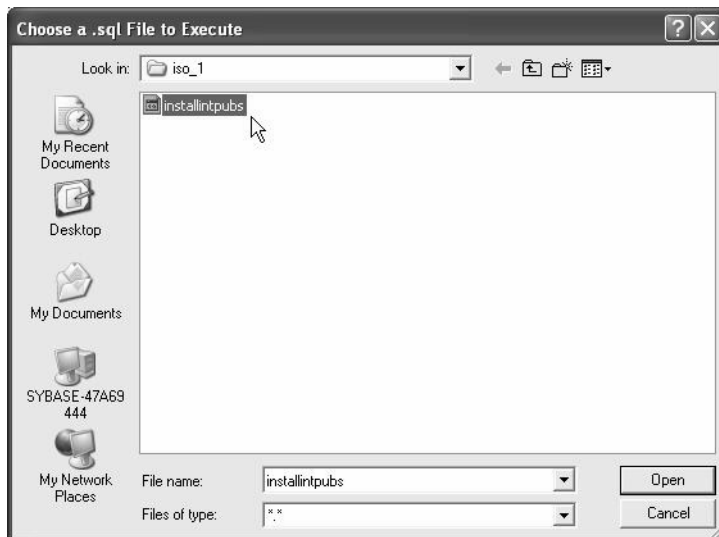
Before you can install the interpubs database using Sybase WorkSpace, you must create a connection profile, as described in “Creating a connection profile for an ASE server” on page 2.

- 1 Launch Sybase WorkSpace.
- 2 Select **Window|Open Perspective|Database Development** from the Sybase WorkSpace main menu bar to open the **Database Development** perspective.
- 3 On the main toolbar, click the **Choose .sql File to Execute** icon.

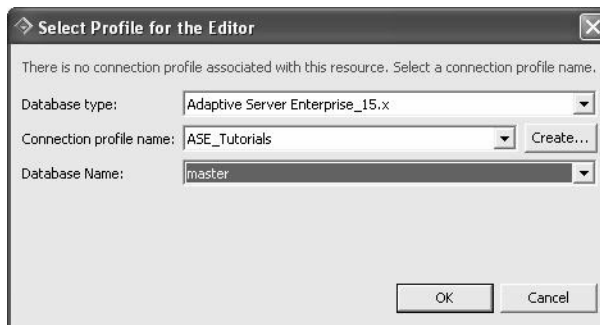


- 4 In the **Choose a .sql File to Execute** dialog box, from the **Files of type** drop-down list, select **\*.\***.
- 5 From the **Look in** drop-down list, navigate to <drive>:  
\`<ASE_install_dir>\scripts\iso_1`.

- 6 Select **installintpubs** and click **Open**.



- 7 In the **Select Profile for the Editor** dialog box, select the following:
- In the **Database type** field, select **Adaptive Server 15.x** or **Adaptive Server 12.5.x**.
  - From the **Connection profile name** drop-down list, select **ASE\_Tutorials**.
  - From the **Database Name** drop-down list, select **master** and click **OK**.



The *installintpubs* script executes and the SQL Results view displays the results.

- 8 In the **Database Explorer** view, right-click **ASE\_Tutorials** and select **Connect** to connect to the server.

The Database Explorer now lists the *interpubs* database under the ASE\_Tutorials connection profile. You are now ready to perform the Adaptive Server Enterprise tutorials. Continue to Chapter 2, “Stored Procedures Tutorial.”

❖ **Installing the *interpubs* database using a command window:**

- 1 To open the command window, on the Windows taskbar, select **Start|Run**.
- 2 Enter `cmd` and then click **OK**.
- 3 Change to `<drive>:\<ASE_install_dir>\scripts\iso_1`.
- 4 Enter `isql -Usa -P -iinstallintpubs`.
- 5 In Sybase WorkSpace, in the **Database Explorer** view, right-click **ASE\_Tutorials** and select **Connect** to connect to the server.

This *interpubs* sample database is now ready for use with this tutorial. You are now ready to perform the Adaptive Server Enterprise tutorials. Continue to Chapter 2, “Stored Procedures Tutorial.”



# Stored Procedures Tutorial

The Stored Procedures tutorial demonstrates how to use Sybase WorkSpace to view, create, and run stored procedures for an Adaptive Server Enterprise database.

A stored procedure is a collection of SQL statements and optional control-of-flow statements stored under a user-specified name. When you create a stored procedure, you can also define parameters to be supplied when the stored procedure is executed.

The ability to write your own stored procedures greatly enhances the power, efficiency, and flexibility of the SQL database language. Because Adaptive Server Enterprise saves the execution plan after running the stored procedures, stored procedures can subsequently run much faster than standalone statements.

For additional background information on the features or concepts for Adaptive Server Enterprise, Sybase WorkSpace, or the Eclipse IDE, see “Related documents” on page v.

## Before you begin

Before you can perform the Stored Procedures tutorials, you must complete some prerequisite installation tasks. See “Setup” on page 1 for step-by-step instructions.

## Working with stored procedures

In this tutorial, you will learn how to use Sybase WorkSpace to view, create, and run stored procedures for an Adaptive Server Enterprise database. This tutorial contains four lessons:

- Lesson 1: View existing stored procedures
- Lesson 2: Create a stored procedure
- Lesson 3: Run a stored procedure
- Lesson 4: View #temp tables during debugging

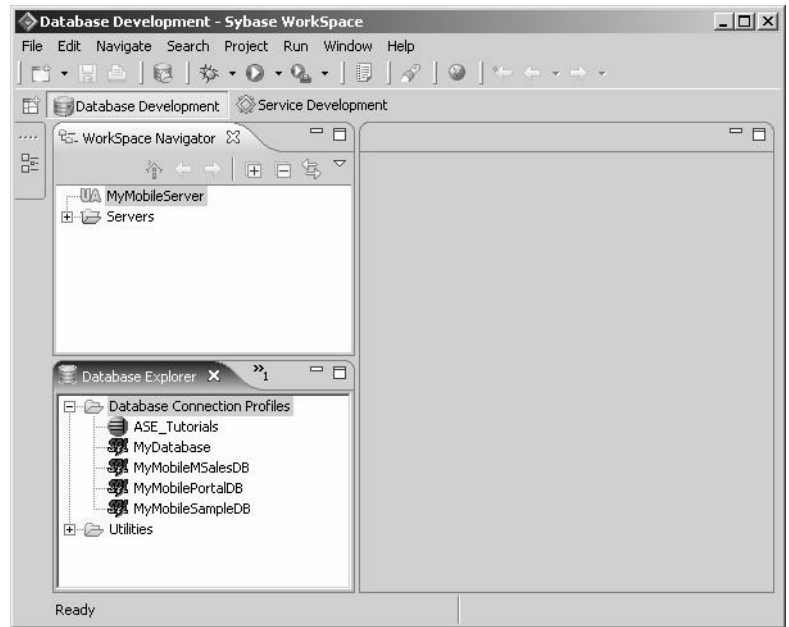
### Lesson 1: View existing stored procedures

In this lesson, you will learn how to use Sybase WorkSpace to view stored procedures for an Adaptive Server Enterprise server. You view and create stored procedures in the Database Development perspective.

- 1 If necessary, open the **Database Development** perspective in Sybase WorkSpace.

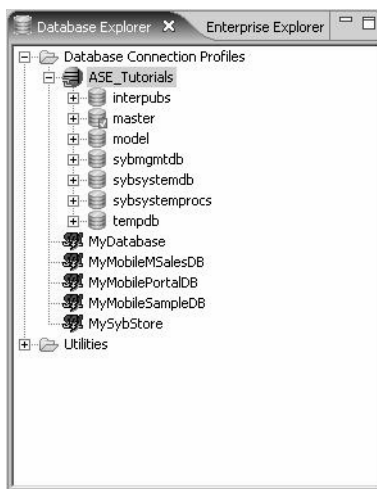
Select **Window|Open Perspective|Database Development** from the Sybase WorkSpace main menu bar.

The Database Development perspective displays.



- 2 In the **Database Explorer** view, expand the contents under the ASE\_Tutorials connection profile.

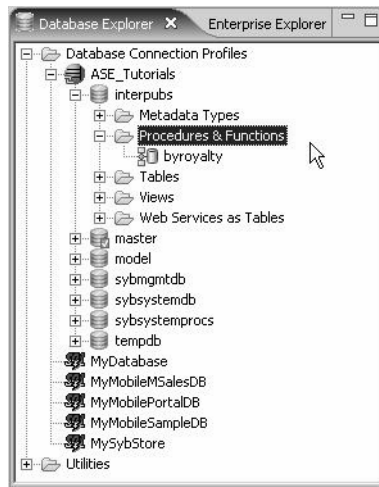
This tutorial uses the *interpubs* database. However, you can also use the *pubs2* or *pubs3* sample database. If you do not see an appropriate sample database under the ASE\_Tutorials connection profile, install one. See “Installing the interpubs database” on page 5.



In **Database Explorer** view, you can view the contents of each database on the server simply by expanding the underlying folders.

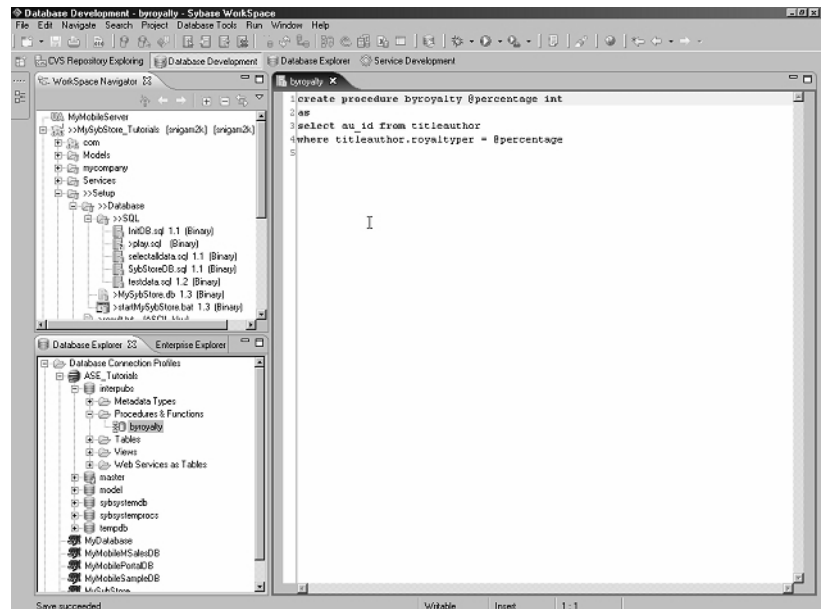
- 3 In the **Database Explorer** view, expand the **Procedures & Functions** folder under the appropriate sample database you are using.

This example uses the international pubs database. You see all the stored procedures contained in this database.



- 4 In the **Database Explorer** view, double-click on any of the stored procedures.

The stored procedure definition displays in the SQL Editor.



If you have appropriate permission for the database connection, you can change the contents of a stored procedure in the SQL Editor and then save it to the server. To save the stored procedure to the server; right-click anywhere in the SQL Editor window and select **Save to Server** from the context menu.

---

**Note** Sybase recommends that you do not change the name of the stored procedure.

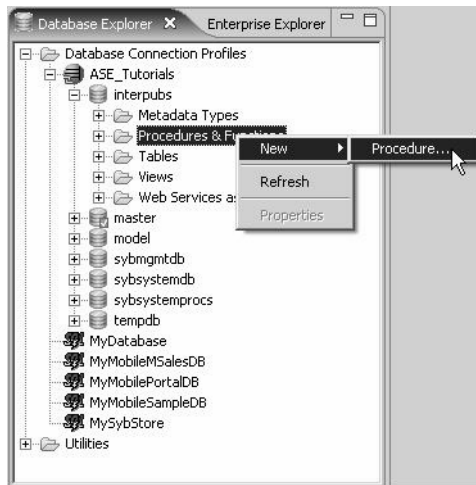
---

- 5 Select **File|Close** on the main menu bar to close the editing pane.

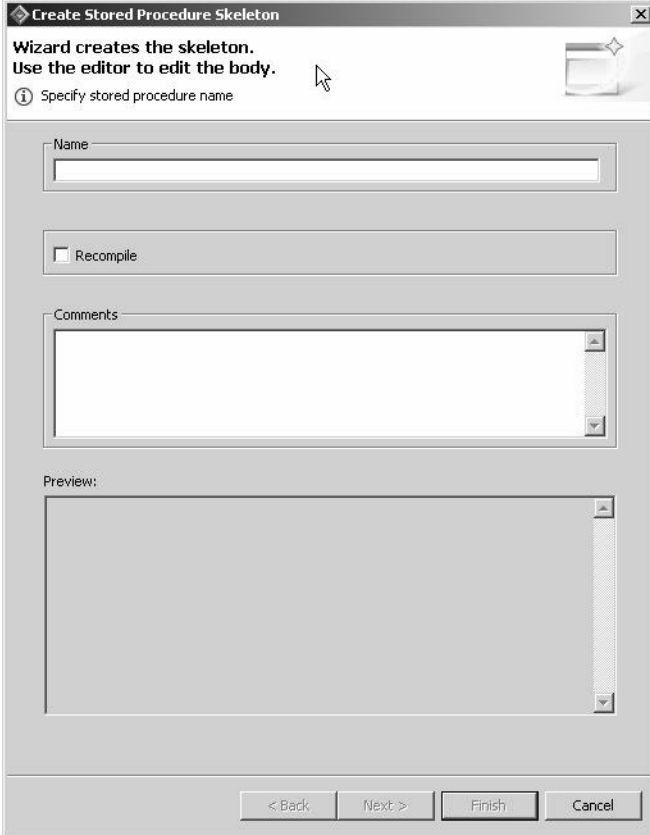
## Lesson 2: Create a stored procedure

In this lesson, you will learn how to use Sybase WorkSpace to create stored procedures for an Adaptive Server Enterprise server database.

- 1 If necessary, select **Window|Open Perspective|Database Development** from the main menu bar to open the **Database Development** perspective.
- 2 In the **Database Explorer** view, right-click the **Procedures & Functions** folder (under the appropriate pubs sample database) and select **New|Procedure**.



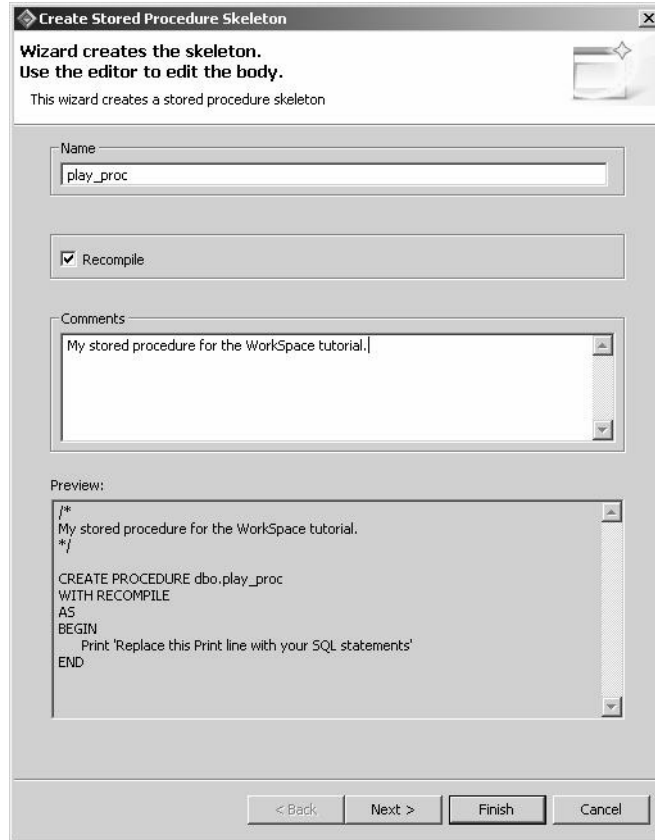
The **Create Stored Procedure Skeleton** wizard displays.



The screenshot shows a dialog box titled "Create Stored Procedure Skeleton". The dialog has a close button (X) in the top right corner. Below the title bar, there is a message: "Wizard creates the skeleton. Use the editor to edit the body." To the right of this message is a small icon of a document with a diamond. Below the message is a help icon (i) and the text "Specify stored procedure name". The main area of the dialog contains a "Name" text box, a "Recompile" checkbox, a "Comments" text area, and a "Preview" text area. At the bottom of the dialog are four buttons: "< Back", "Next >", "Finish", and "Cancel".

- 3 In the **Name** field, enter `play_proc`.

Optionally, you can click the **Recompile** check box or add comments. As you enter or change information in this wizard, its automatically updates the preview.



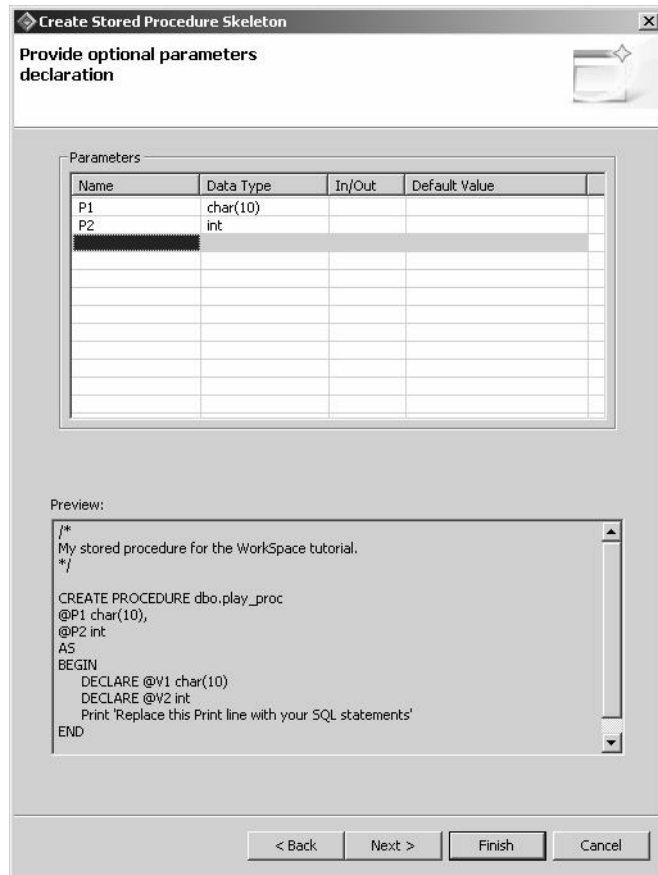
Click **Next** to continue.

- 4 Enter the parameters for this stored procedure.

On the parameters page, enter the following two parameters. You can manually enter the data type or select it from the drop-down list.

- P1 – with a data type of char(10). Although char(10) does not appear on the drop-down list, you can manually enter it in the Data Type field.
- P2 – with a data type of int.

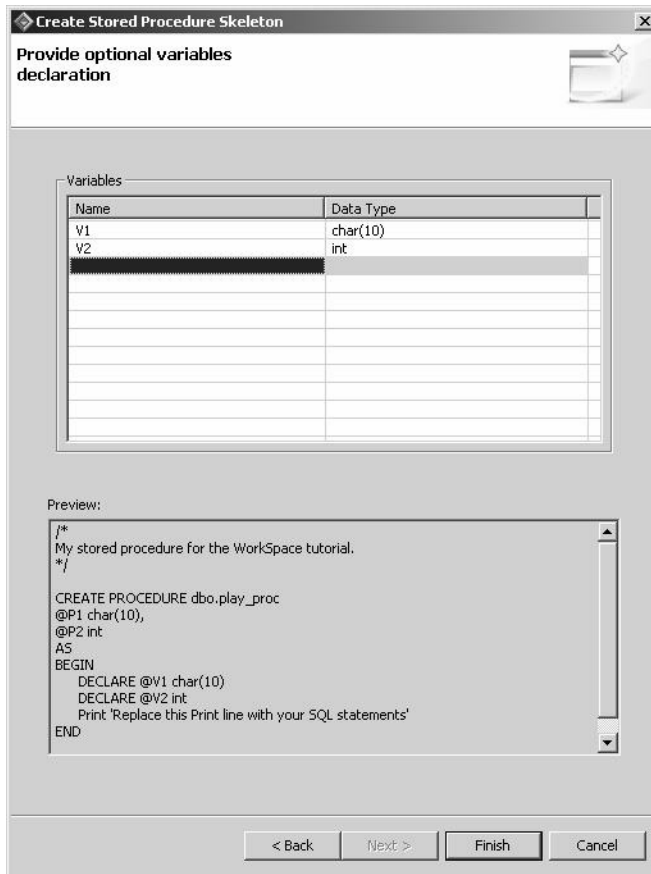




- 5 Click **Next** to enter variables for this stored procedure.

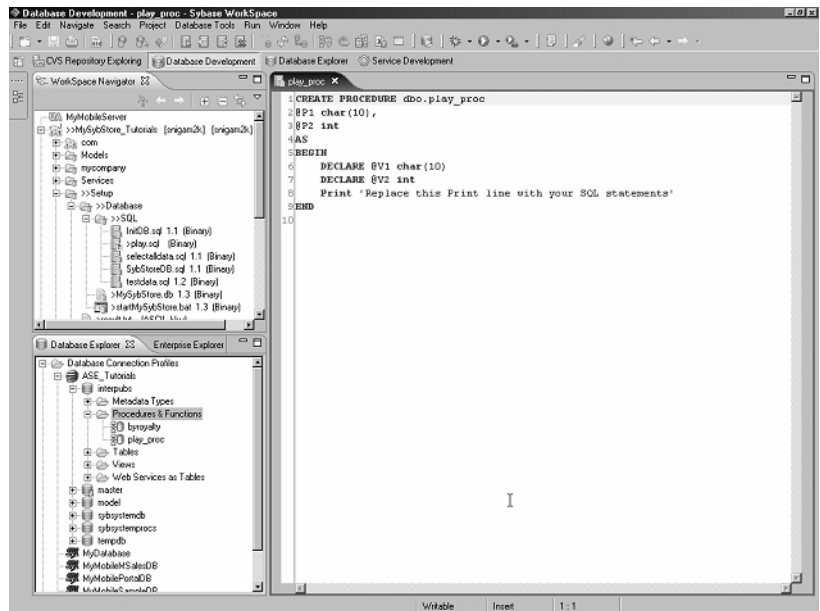
You can manually enter the data type or select it from the drop-down list.

- V1 – with a data type of char(10). Although char(10) does not appear on the drop-down list, you can manually enter it in the Data Type field.
- V2 – with a data type of int.



6 Click **Finish**.

The wizard creates a skeleton of the stored procedure in the database and displays it in the SQL Editor.



We will now add some basic SQL code to demonstrate some Sybase WorkSpace features as well as add a #temp table, which you will view and modify in a later lesson.

- 7 Add the following SQL statements to the stored procedure after the Print statement.

```

create table #play_table
(c1 char(10),
c2 int)

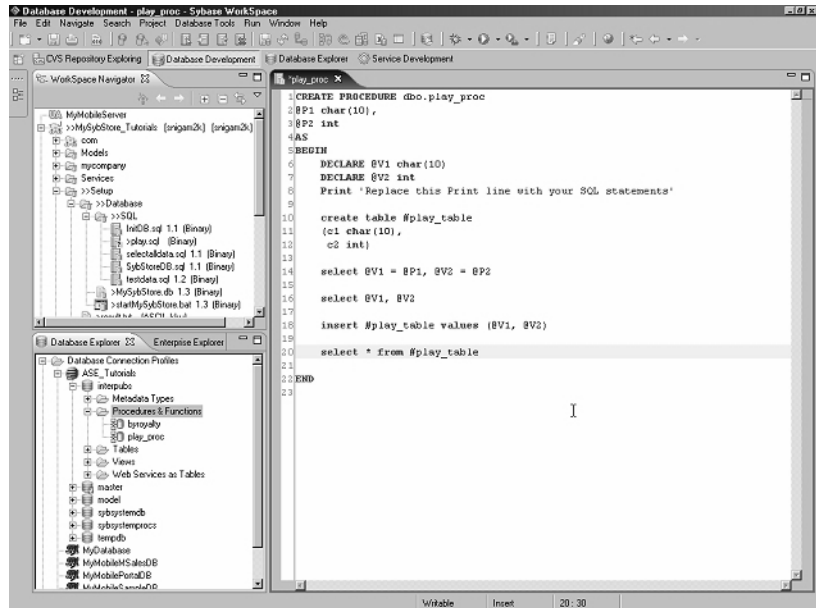
select @V1 = @P1, @V2 = @P2

select @V1, @V2

insert #play_table values (@V1, @V2)

select * from #play_table

```



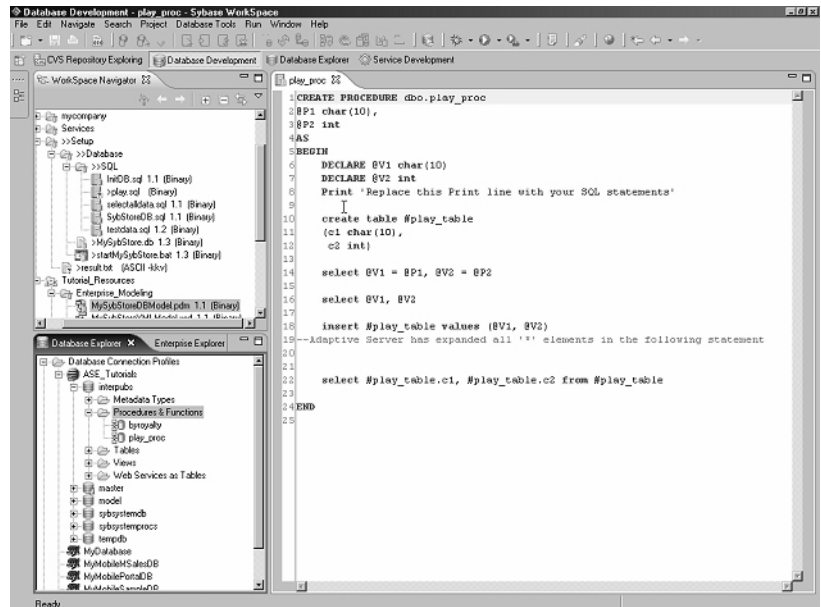
- 8 Save the stored procedure to the server; right-click anywhere in the SQL Editor window and select **Save to Server** from the context menu.

You will use this stored procedure in subsequent tutorials.

- 9 Look at the text at the bottom of the SQL Editor.

Notice how Sybase WorkSpace expands the \* (asterisk) into the complete column list from the table.

- Now look in the **Database Explorer** view under the **Procedures & Functions** folder to see that Sybase WorkSpace added the new stored procedure, `play_proc`, to the list.



- Select **File|Close** on the main menu bar to close the editing pane.

You can easily make changes to the stored procedure and save it back to the server.

## Lesson 3: Run a stored procedure

In this lesson, you will learn how to use Sybase WorkSpace to run (execute) a stored procedure for an Adaptive Server Enterprise server database.

**Note** Before you can complete this tutorial, you must create the `play_proc` stored procedure, as described in the previous lesson, “Lesson 2: Create a stored procedure” on page 14.

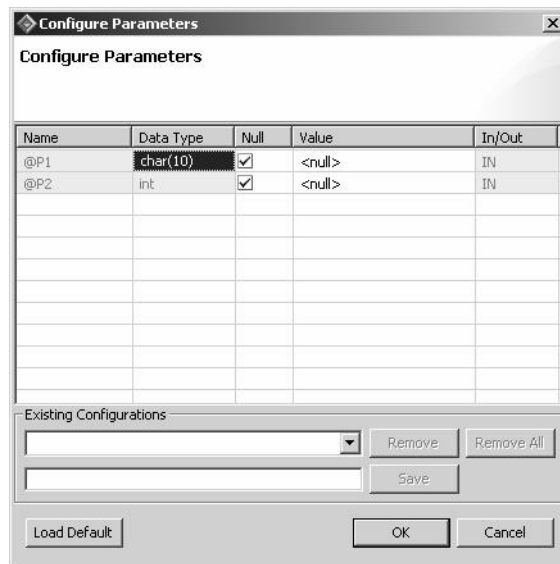
- If necessary, select **Window|Open Perspective|Database Development** from the main menu bar to open the **Database Development** perspective.

- 2 In the **Database Explorer** view, open the **Procedures & Functions** folder under the appropriate sample database.

Look for the play\_proc stored procedure that you created in the previous tutorial, “Lesson 2: Create a stored procedure” on page 14. You can run the stored procedure in one of two ways:

- Double-click this stored procedure to open it in the editing pane. Right-click in the editing pane and select **Run** from the context menu.
- Right-click this procedure and select **Run Procedure** from the context menu.

Once you run the stored procedure, the **Configure Parameters** dialog box displays.



Next, enter the values for the stored procedure invocation.

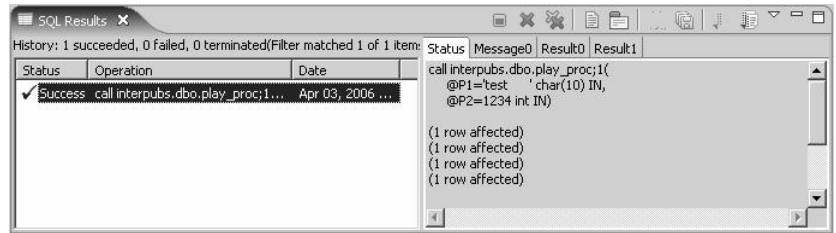
- 3 Enter any values you like for the stored procedure invocation.

Enter values the first time you run a stored procedure. On subsequent runs, you can use the values you previously entered or enter new values.

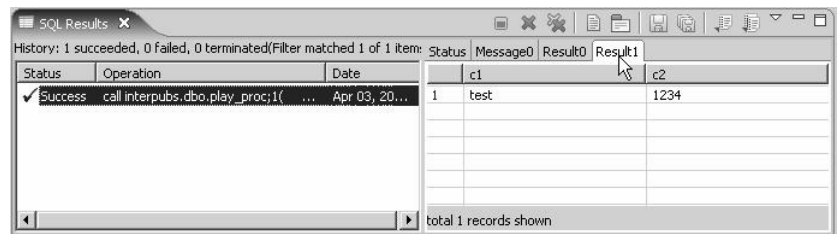
- 4 Click **OK** to run the procedure.

Sybase WorkSpace executes the stored procedure and places the results in the **SQL Results** view.

Click the **Result History** line in the left pane of the **SQL Results** view to display the results in the right pane.



- Review the different tables and results that the stored procedure returns in the the **SQL Results** view.



- Select **File|Close** on the main menu bar to close the SQL Editor.

You can easily make changes to the stored procedure and save it back to the server.

## Lesson 4: View #temp tables during debugging

In this lesson, you will learn how to use Sybase WorkSpace to view and change the contents of the referenced tables for an Adaptive Server Enterprise server database.

Before you can perform the steps in this lesson, create a stored procedure and then run the stored procedure, as described in the previous lessons, “Lesson 2: Create a stored procedure” on page 14 and “Lesson 3: Run a stored procedure” on page 21.

### ❖ Viewing #temp tables during debugging

- If necessary, select **Window|Open Perspective|Database Development** from the main menu bar to open the **Database Development** perspective.

- 2 In the **Database Explorer** view, open the **Procedures & Functions** folder under the appropriate pubs sample database.
- 3 Find the play\_proc stored procedure that you created in “Lesson 2: Create a stored procedure” on page 14 and double-click it to open it in the SQL Editor.

To view the contents of a #temp table for a stored procedure, use the stored procedure debugger. When you stop at a breakpoint in the stored procedure, you can view and modify the contents of the #temp table.

First, add breakpoints to the stored procedure and then invoke the debugger.

- 4 Add breakpoints to the following lines.

If you look at the following figure, these statements correspond to line numbers 10, 14, 16, 18, and 22. However, if you added comments or selected the Recompile option when creating the stored procedure, your line numbers may differ.

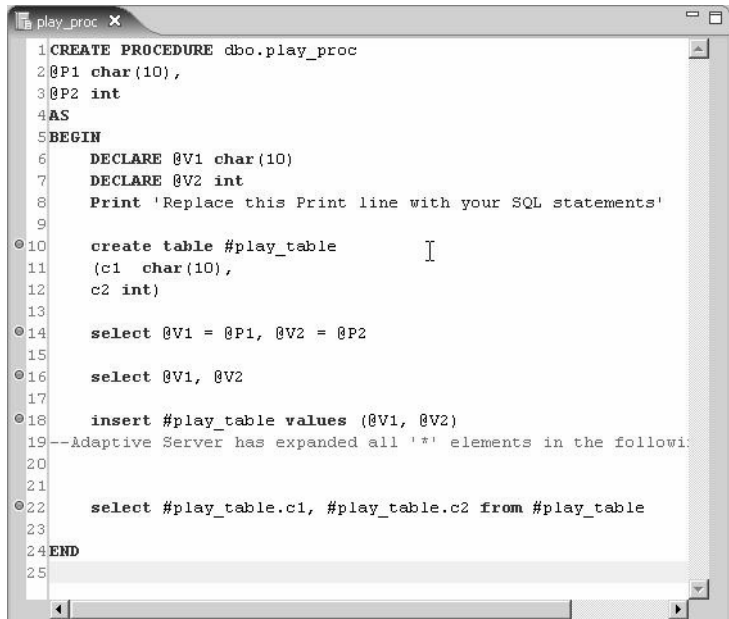
```
Create table #play_table
select @V1 = @P1, @V2 = @P2
select @V1, @V2
insert #play_table values (@V1, @V2)
select #play_table.c1, #play_table.c2 from #play_table
```

You can add a breakpoint using either method:

- Double-click the vertical gray border on the immediate left of the line number.
- Right-click the vertical gray border to the immediate left of the line number and select **Toggle Breakpoint**.



Sybase WorkSpace indicates a breakpoint by a light blue dot to the left of the line number in the SQL Editor.

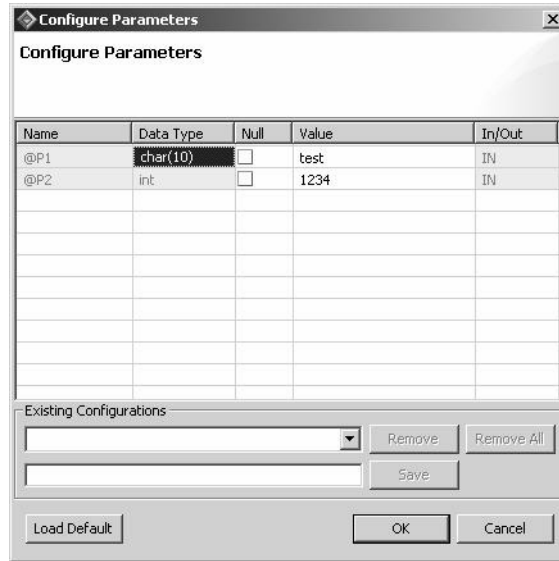


```
1 CREATE PROCEDURE dbo.play_proc
2 @P1 char(10),
3 @P2 int
4 AS
5 BEGIN
6     DECLARE @V1 char(10)
7     DECLARE @V2 int
8     Print 'Replace this Print line with your SQL statements'
9
10    create table #play_table
11    (c1 char(10),
12     c2 int)
13
14    select @V1 = @P1, @V2 = @P2
15
16    select @V1, @V2
17
18    insert #play_table values (@V1, @V2)
19 --Adaptive Server has expanded all '*' elements in the following
20
21
22    select #play_table.c1, #play_table.c2 from #play_table
23
24 END
25
```

Now you will debug the stored procedure.

- 5 Right-click in the SQL Editor and select **Debug** from the context menu.

The Configure Parameters wizard displays.



- 6 Next, retain or change the values for the stored procedure invocation.

As you can see, the Value fields display the values you selected when initially running the stored procedure.

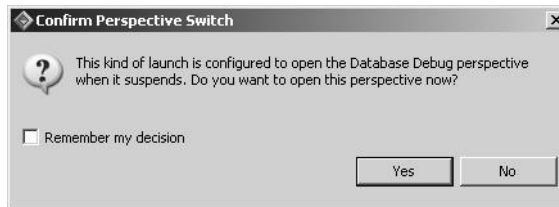
---

**Note** Enter values the first time you run/debug a stored procedure. When subsequently running the stored procedure, you can use the values you previously entered or enter new values.

---

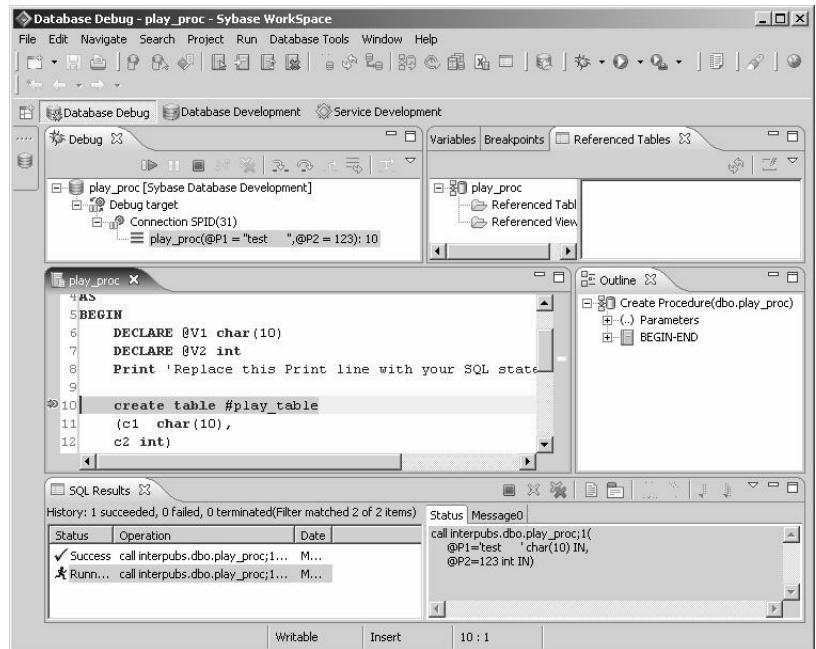
- 7 Click **OK** to run the procedure.

The stored procedure execution starts, and Confirm Perspective Switch dialog box displays, which confirms that you want to change to the Database Debug perspective.



- 8 Select **Yes** in the **Confirm Perspective Switch** box to switch to the **Database Debug** perspective.

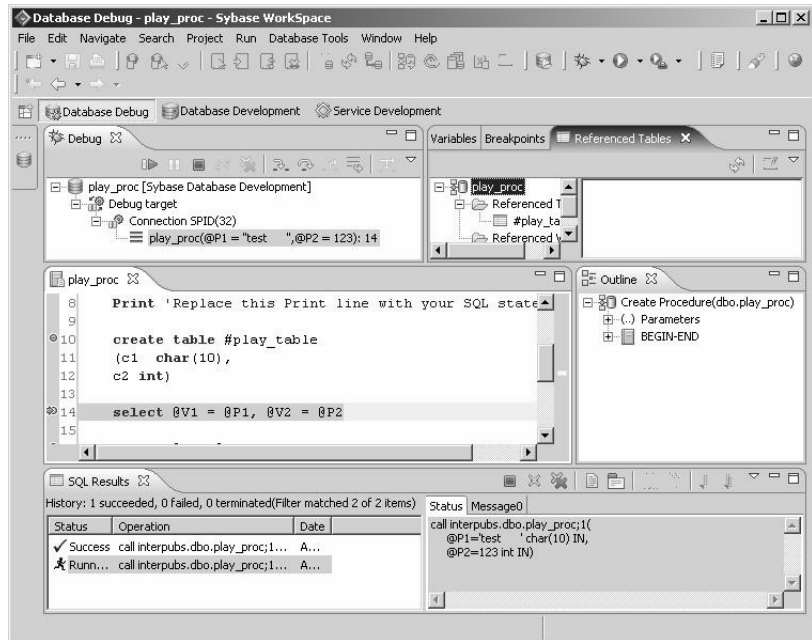
Sybase WorkSpace stops at the first breakpoint on line 10.



- 9 Click the **Resume** icon once, located on the **Debug** view toolbar, to step to the next breakpoint.

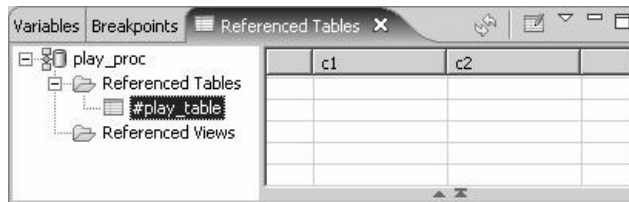


The Referenced Tables view opens in the upper right corner of the Database Debug perspective.



- 10 Click the #play\_table temp table in the **Referenced Tables** view to display its contents.

At this point the temp table is empty because you just created it.

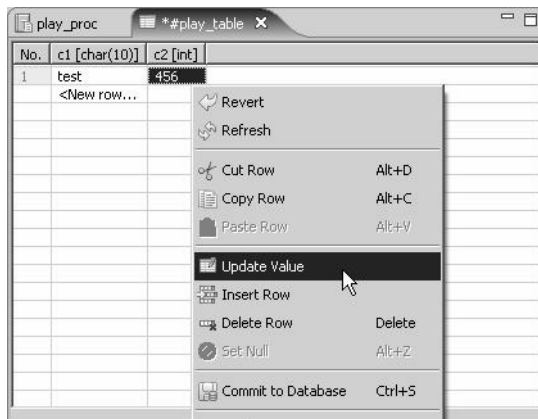


- 11 Finish walking through the breakpoints until you reach line 22 (just below the insert statement) by clicking the **Resume** icon three more times.

Now, you can view and modify the contents of the #play\_table.



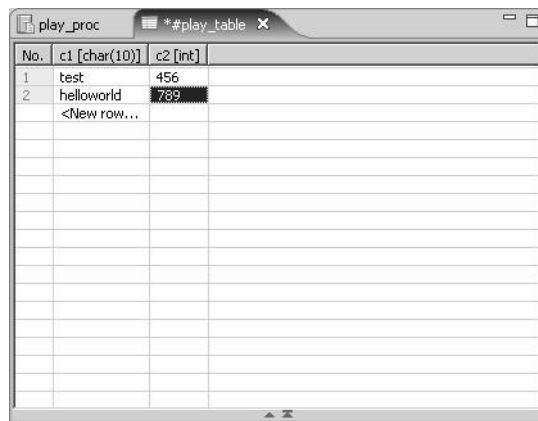
Select **123** in the **C2[int]** column, right-click, and select **Update Value** from the context menu.



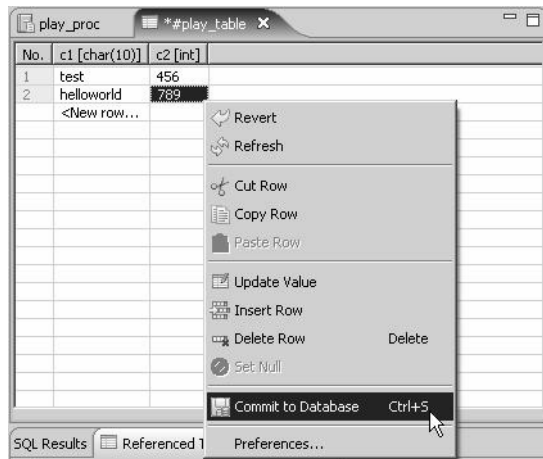
Sybase WorkSpace highlights the existing value.

- 15 Enter 456 to overwrite the existing value.
- 16 Insert another row in the table using either method:
  - Double-click **<New row>**.
  - Right-click anywhere in the #play\_table grid and select **Insert Row** from the context menu.

Add some new values for the new row.



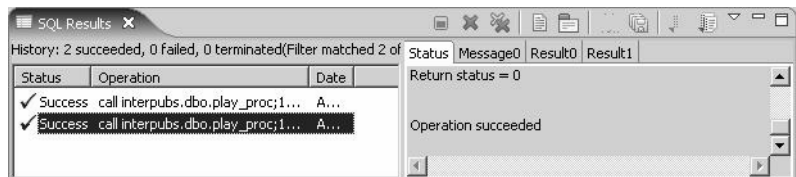
- 17 Right-click anywhere in the #play\_table grid and select **Commit to Database** from the context menu.



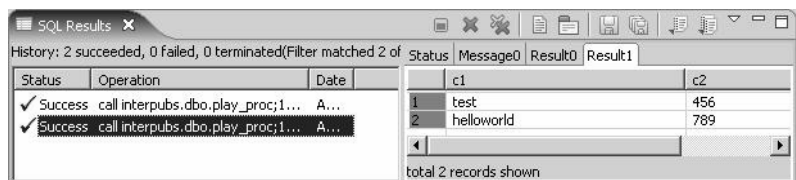
The #play\_table contents you just changed in the table editor now match the table contents in the Referenced Tables view.

- 18 Continue execution of the stored procedure and confirm that the changes you made to the #play\_table display in the **SQL Results** view.

Display the play\_proc editor by clicking its tab. Complete the stored procedure execution by selecting the **Resume** icon one more time.



- 19 Click the **Result1** tab in the **SQL Results** view to see the values reflect the modifications that you made during the debug session.



20 Select **File|Close** on the main menu bar to close the editor.

You can easily make changes to the stored procedure and save it back to the server.



# Triggers Tutorials

These tutorials demonstrate how to use Sybase WorkSpace to view, create, and debug triggers on an Adaptive Server Enterprise server.

A trigger is a SQL procedure that initiates an action when an event (insert, delete, or update) occurs. Because triggers are event-driven specialized procedures, they are stored in and managed by Adaptive Server Enterprise. Adaptive Server Enterprise automatically fires the trigger as a result of a data modification to the associated table.

A table can have triggers for *insert*, *delete*, and *update*, or a combination of these three event types.

For additional background information on the features or concepts for Adaptive Server Enterprise, Sybase WorkSpace, or the Eclipse IDE, see “Related documents” on page v.

## Before you begin

Before you can perform the Triggers tutorials, you must complete some prerequisite installation tasks. See “Setup” on page 1 for step-by-step instructions.

## Viewing and creating triggers

In the following lessons, you will learn how to use Sybase WorkSpace to view an existing trigger and create a new trigger for an Adaptive Server Enterprise server.

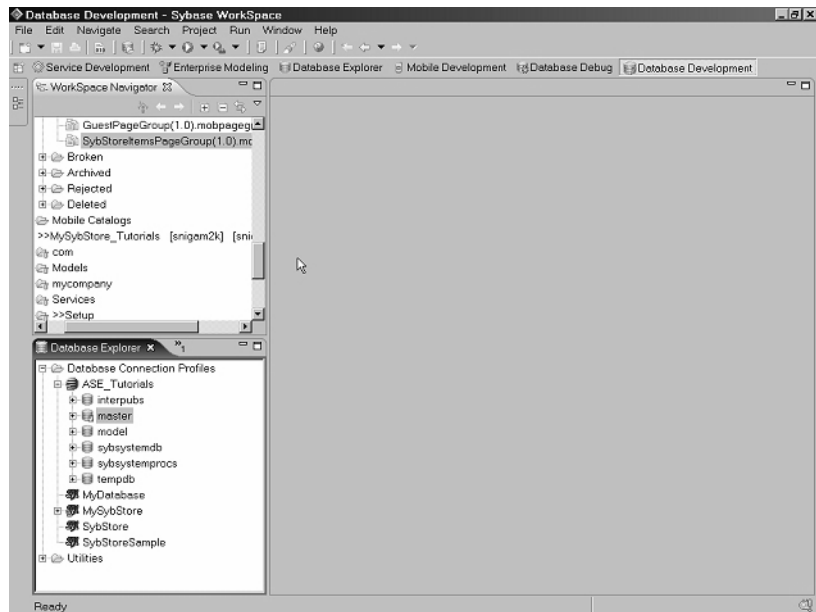
Lesson 1: View an existing trigger

Lesson 2: Create a new trigger

## Lesson 1: View an existing trigger

In this lesson, you will use Sybase WorkSpace to view an existing trigger in the *titles* table in the *interpubs* database.

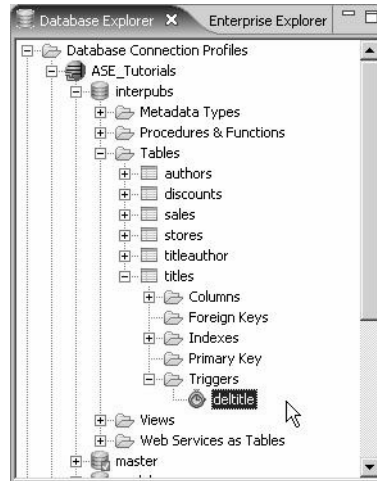
- 1 In Sybase WorkSpace, select **Window|Open Perspective|Database Development** from the main menu bar to open the **Database Development** perspective.
- 2 In the **Database Explorer** view, expand the contents under the ASE\_Tutorials connection profile that you created during the setup process.



In the **Database Explorer** view, you can view the contents of each database on the server by expanding the underlying folders.

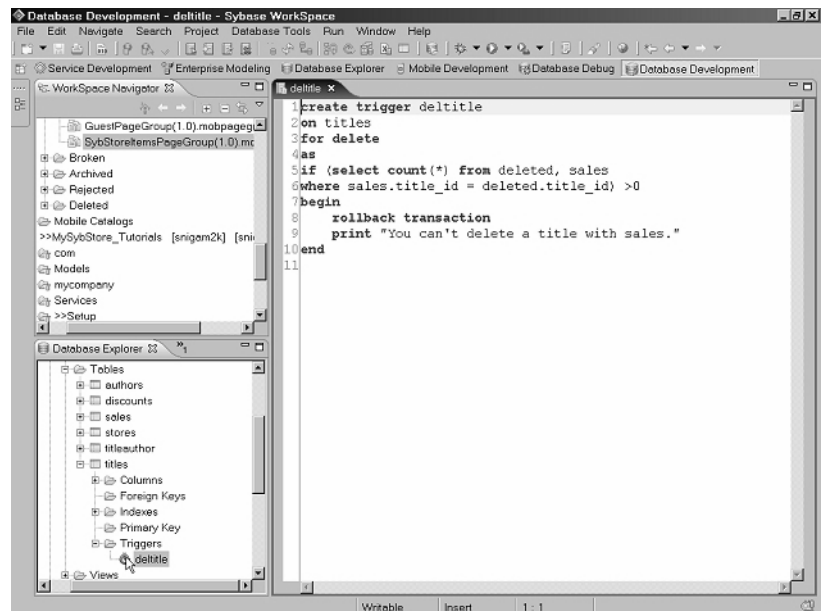
- 3 In the **Database Explorer** view, expand the **Tables** folder under the sample pubs database that you installed during the tutorial setup process.

Triggers reside with their corresponding table. Expand the **titles** table folder and the **Triggers** folder under that. Look for the **deltitle** trigger.



4 In the **Database Explorer** view, double-click the **deltitle** trigger.

The SQL Editor displays the trigger code.



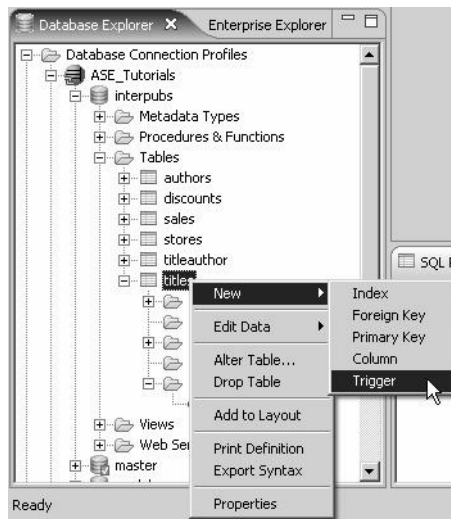
You can change the code of any trigger, then save it to the server by right-clicking anywhere in the SQL Editor window and selecting **Save to Server** from the context menu.

- 5 Select **File|Close** on the main menu bar to close the SQL Editor.

## Lesson 2: Create a new trigger

In this lesson, you will learn how to use Sybase WorkSpace to create a new trigger for an Adaptive Server Enterprise server database table.

- 1 If necessary, select **Window|Open Perspective|Database Development** from the main menu bar to open the **Database Development** perspective.
- 2 In the **Database Explorer** view, expand the **Tables** folder under the sample pubs database that you installed during the tutorial setup process. Triggers reside with their corresponding table.
- 3 In the **Database Explorer** view, right-click the **titles** table and select **New|Trigger**.



The **Create Trigger Skeleton** wizard displays.

- 4 In the **Name** field, enter `insert_title`.
- 5 Click the **For insert** check box.

Optionally, you can add a comment. Sybase WorkSpace updates the trigger code in the Preview box as you enter data.

**Create Trigger Skeleton**

Wizard creates the skeleton.  
Use the editor to edit the body.

Name  
insert\_title

Table  
titles

For update  For insert  For delete

Comments  
insert trigger for titles table

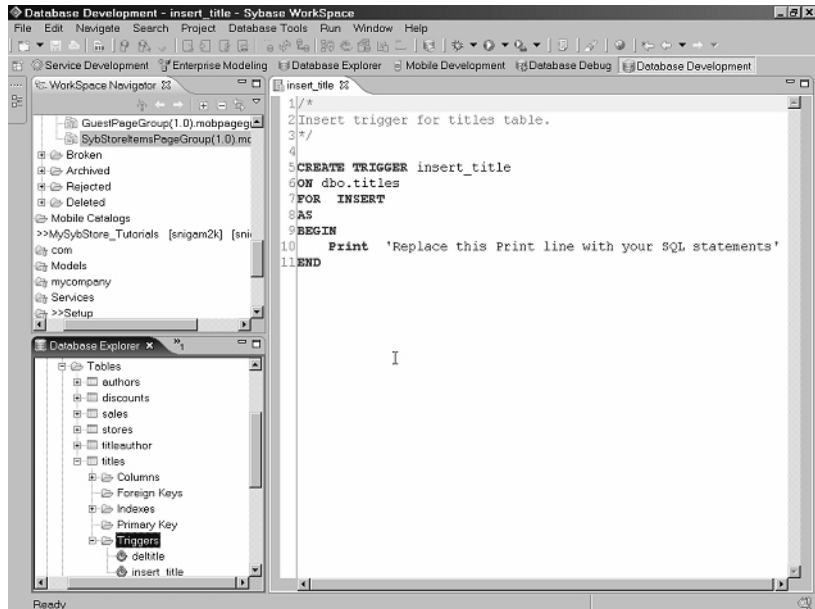
Preview:

```
/*  
insert trigger for titles table  
*/  
  
CREATE TRIGGER insert_title  
ON dbo.titles  
FOR INSERT  
AS  
BEGIN  
    Print 'Replace this Print line with your SQL statements'  
END
```

Finish Cancel

6 Click **Finish**.

The SQL Editor creates and displays a skeleton of the trigger so that you can finish it and then save it to the server.



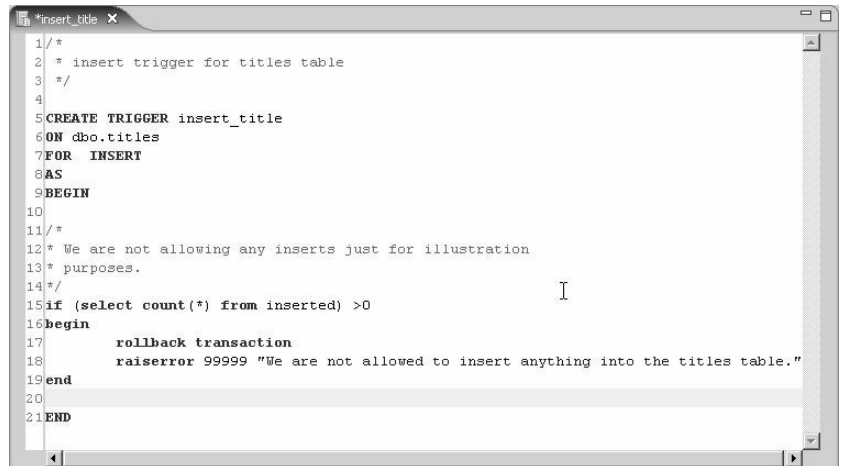
- 7 Add the following SQL statements to the trigger after the **begin** statement.

```

/*
 * We are not allowing any inserts just for
 * illustration purposes.
 */
if (select count(*) from inserted) >0
begin
    rollback transaction
    raiserror 99999 "We are not allowed to insert
anything into the titles table."
end

```

In this example, the SQL code indicates to display an error message if any user tries to insert data into the titles table. Format your code in the SQL Editor to look like the following example.



```
*insert_title x
1/*
2 * insert trigger for titles table
3 */
4
5CREATE TRIGGER insert_title
6ON dbo.titles
7FOR INSERT
8AS
9BEGIN
10
11/*
12 * We are not allowing any inserts just for illustration
13 * purposes.
14 */
15if (select count(*) from inserted) >0
16begin
17    rollback transaction
18    raiserror 99999 "We are not allowed to insert anything into the titles table."
19end
20
21END
```

- 8 Right-click anywhere in the SQL Editor window and select **Save to Server** to save your changes.

You will use this trigger in subsequent tutorials.

- 9 Select **File|Close** on the main menu bar to close the SQL Editor.

You can easily make changes to the trigger and save it back to the server.

## Running and debugging a trigger

In the following lessons, you will learn how to use Sybase WorkSpace to run (execute) a trigger, view the inserted special table for a trigger, and debug a trigger for a database table in an Adaptive Server Enterprise server.

Lesson 1: Run a trigger

Lesson 2: View the inserted special table within a trigger

Lesson 3: Debug the trigger

---

**Note** Before you can start these lessons, you must create the *insert\_title* trigger, as described in the previous tutorial, “Lesson 2: Create a new trigger” on page 36.

---

### Lesson 1: Run a trigger

In this lesson, you will use Sybase WorkSpace to run the trigger you just created in the *titles* table.

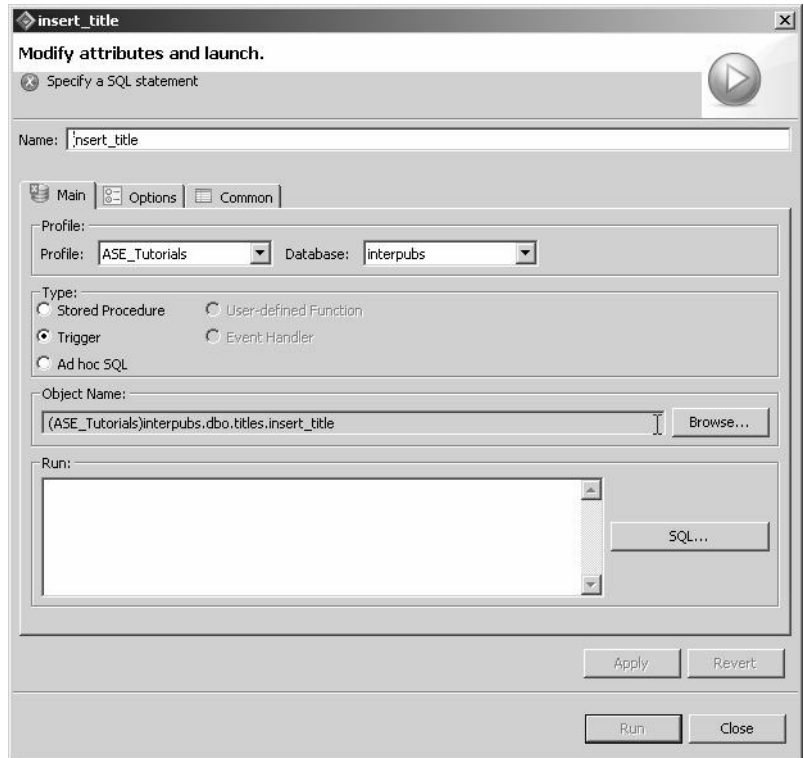
- 1 If necessary, select **Window|Open Perspective|Database Development** from the main menu bar to open the **Database Development** perspective.
- 2 Find the *insert\_title* trigger that you created in the previous tutorial, “Lesson 2: Create a new trigger” on page 36.

In the **Database Explorer** view, expand the **Tables** folder under the appropriate pubs sample database. Next, expand the **titles** folder and then the **Triggers** folder.

- 3 Run the trigger using either method:
  - Double-click the trigger to open it in the SQL Editor, and then right-click anywhere in the SQL Editor and select **Run** from the context menu.
  - Right-click the trigger and select **Run Trigger** from the context menu.



The **Modify attributes and launch** dialog box displays.



- 4 In the **Run** box, enter the following SQL statement that executes to cause the trigger invocation.

```

INSERT INTO titles
    ( title_id,
      title,
      type,
      price,
      advance,
      royalty,
      ytd_sales,
      notes,
      pubdate )
VALUES ( 'XX9999',
        'This Book Cannot Be Inserted',
        'drama',
        19.22,
        10000.00,

```

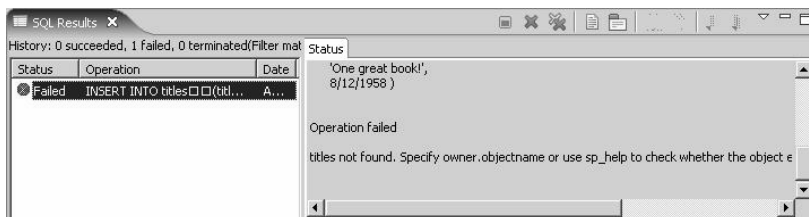
```
10,  
99,  
'One great book! ',  
'8/12/1958' )
```

Although you can click **SQL** to launch the Visual SQL dialog to create the SQL statement, for this tutorial, manually enter the SQL statement in the Run box.

- 5 Click **Apply** to save the SQL statement for future invocations.
- 6 Click **Run** to execute the trigger.

Sybase WorkSpace executes the triggers and displays the results in the SQL Results view. The insert failed as expected because the trigger was designed to print an error message during an attempt to insert data into the table.

The **Status** tab displays the message that the raiserror command issued.



- 7 Select **File|Close** on the main menu bar to close the SQL Editor.

You can easily make changes to the trigger and save it back to the server.

## Lesson 2: View the inserted special table within a trigger

During the debugging process, one way to ensure that data modifications are consistent throughout all tables in a database is called referential integrity. To manage referential integrity, you can create triggers that take effect when you give insert, update, and delete commands for particular tables or columns.

You can view the contents of the *inserted* and *deleted* special tables, which contain data that the trigger inserts, deletes, or updates (delete followed by insert). These special tables are temporary and exist only in memory during trigger execution. Once the trigger stops running, these tables no longer exist. After debugging, you would want to remove or comment out these `Select *` from statements.

In this lesson, you will learn how to view and change contents of the sample **titles** table, and how to query the contents of the *inserted* special trigger table so that you can view its contents in the SQL Results view.

---

**Note** Before you can perform this tutorial, you must create the *insert\_title* trigger and run it, as described above.

---

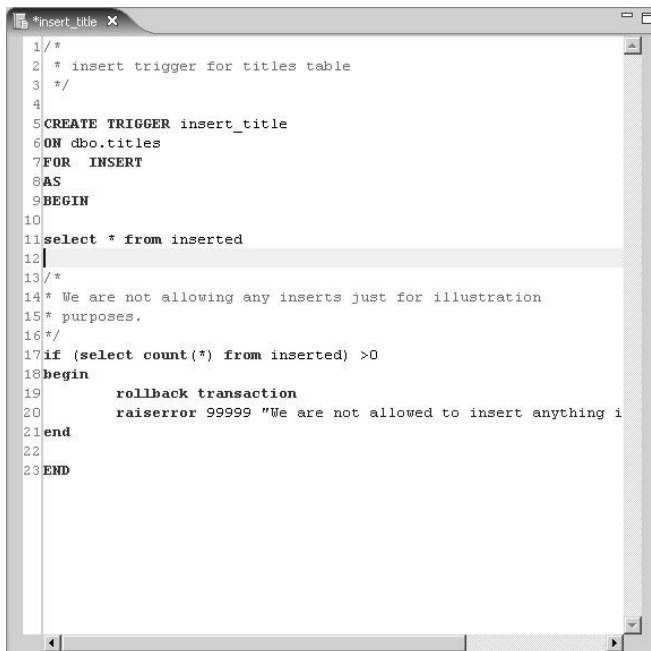
- 1 If necessary, select **Window|Open Perspective|Database Development** from the main menu bar to open the Database Development perspective.
- 2 Find the *insert\_title* trigger that you created in “Lesson 2: Create a new trigger” on page 36.

In the **Database Explorer** view, expand the **Tables** folder under the appropriate pubs sample database. Next, expand the **titles** folder and then the **Triggers** folder.

- 3 Double-click the *insert\_title* trigger to open it in the SQL Editor.  
Next, add select statements to the trigger that return the contents of the inserted rows in the SQL Results view.
- 4 Add the following SQL statement to the trigger after the BEGIN command but before the comments (line 11).

```
select * from inserted
```

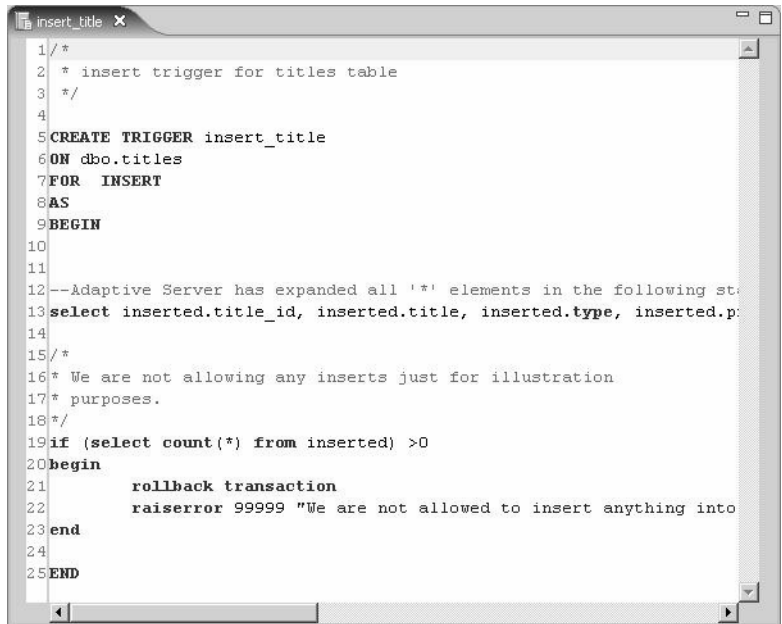
Format your code in the SQL Editor to look like this.



```
1/*
2 * insert trigger for titles table
3 */
4
5CREATE TRIGGER insert_title
6ON dbo.titles
7FOR INSERT
8AS
9BEGIN
10
11select * from inserted
12|
13/*
14 * We are not allowing any inserts just for illustration
15 * purposes.
16 */
17if (select count(*) from inserted) >0
18begin
19     rollback transaction
20     raiserror 99999 "We are not allowed to insert anything i
21end
22
23END
```

- 5 Right-click anywhere in the SQL Editor window and select **Save to Server** to save the changes.

The server expands the \* (asterisks) contained in the **select** statement.



```

1 /*
2  * insert trigger for titles table
3  */
4
5 CREATE TRIGGER insert_title
6 ON dbo.titles
7 FOR INSERT
8 AS
9 BEGIN
10
11
12 --Adaptive Server has expanded all '*' elements in the following st
13 select inserted.title_id, inserted.title, inserted.type, inserted.p
14
15 /*
16 * We are not allowing any inserts just for illustration
17 * purposes.
18 */
19 if (select count(*) from inserted) >0
20 begin
21     rollback transaction
22     raiserror 99999 "We are not allowed to insert anything into
23 end
24
25 END

```

Next, add breakpoints to prepare for debugging the trigger.

### Lesson 3: Debug the trigger

In this lesson, you will debug your trigger. Before you can do so, you must create the *insert\_title* trigger and run it.

- 1 If necessary, click to open the *insert\_title* trigger in the SQL Editor.
- 2 Add breakpoints to lines 19, 21, and 22 using either method:
  - Double-click the vertical gray border on the immediate left of the line number.
  - Right-click the vertical gray border to the immediate left of the line number and select **Toggle Breakpoint**.

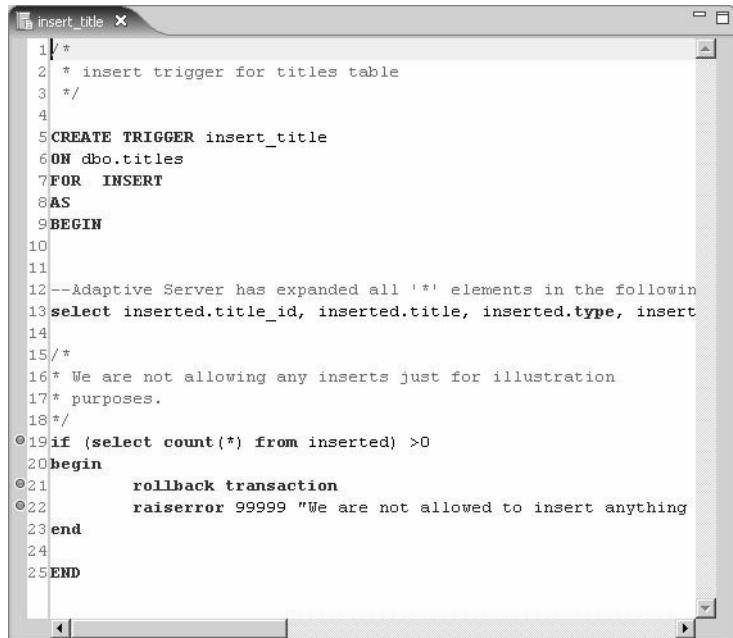
In case your line numbers differ, add breakpoints to the following lines.

```
if (select count(*) from inserted)>0
```

```
rollback transaction
```

```
raiserror 99999 "We are not allowed to insert  
anything into the titles table."
```

Sybase WorkSpace indicates a breakpoint by a light blue dot next to the line number in the SQL Editor.



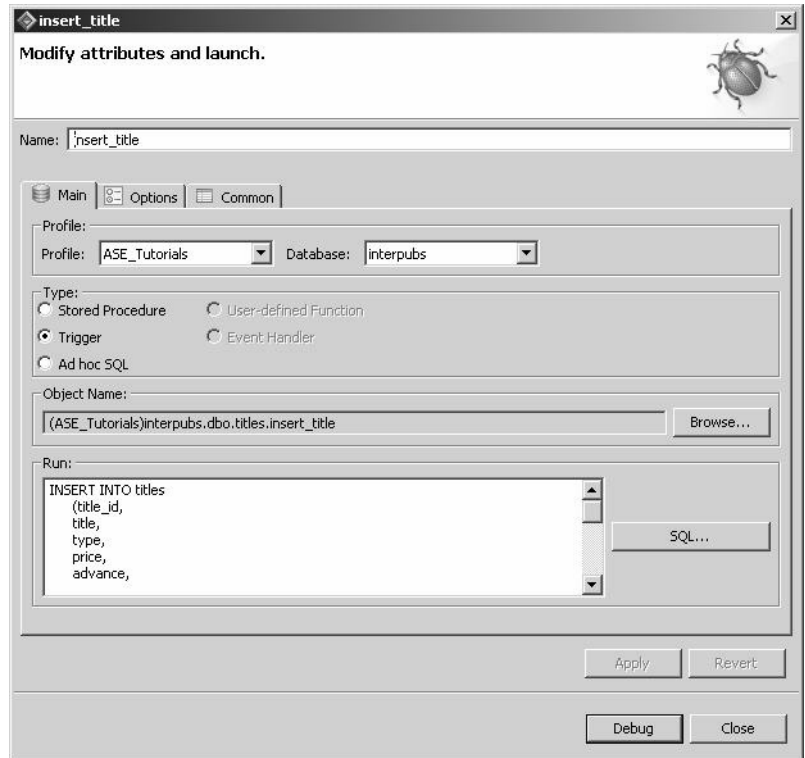
The screenshot shows a window titled 'insert\_title' containing SQL code for a trigger. The code is as follows:

```
1 /*  
2 * insert trigger for titles table  
3 */  
4  
5 CREATE TRIGGER insert_title  
6 ON dbo.titles  
7 FOR INSERT  
8 AS  
9 BEGIN  
10  
11  
12 --Adaptive Server has expanded all '*' elements in the following  
13 select inserted.title_id, inserted.title, inserted.type, insert  
14  
15 /*  
16 * We are not allowing any inserts just for illustration  
17 * purposes.  
18 */  
19 if (select count(*) from inserted) >0  
20 begin  
21     rollback transaction  
22     raiserror 99999 "We are not allowed to insert anything  
23 end  
24  
25 END
```

Light blue dots (breakpoints) are placed next to line numbers 19, 21, and 22.

- 3 To debug the trigger; right-click anywhere in the SQL Editor and select **Debug** from the context menu.

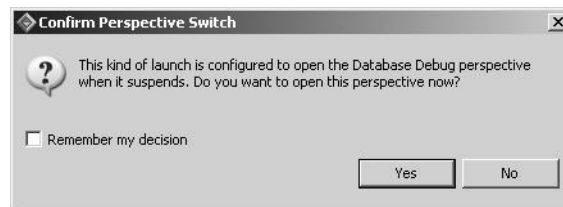
The **Modify attributes and launch** dialog box displays.



Sybase WorkSpace displays the values you previously entered in “Running and debugging a trigger” on page 40.

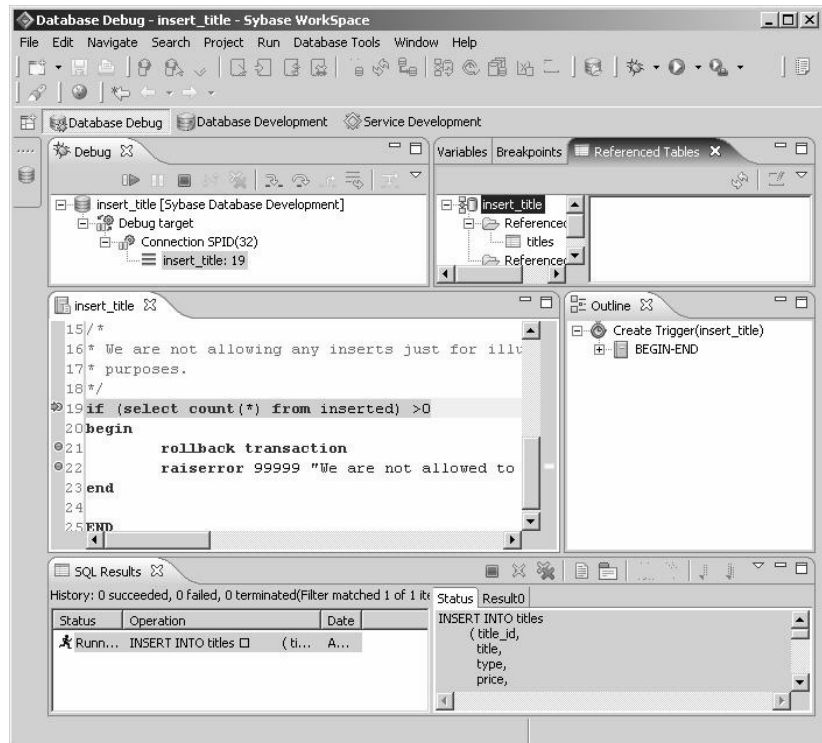
- 4 Click **Debug** to continue.

Execution starts and the Confirm Perspective Switch dialog box displays, which confirms that you want to change to the Database Debug perspective.



- 5 Select **Yes** in the **Confirm Perspective Switch** box to switch to the **Database Debug** perspective.

Sybase WorkSpace stops at the first breakpoint on line 19.



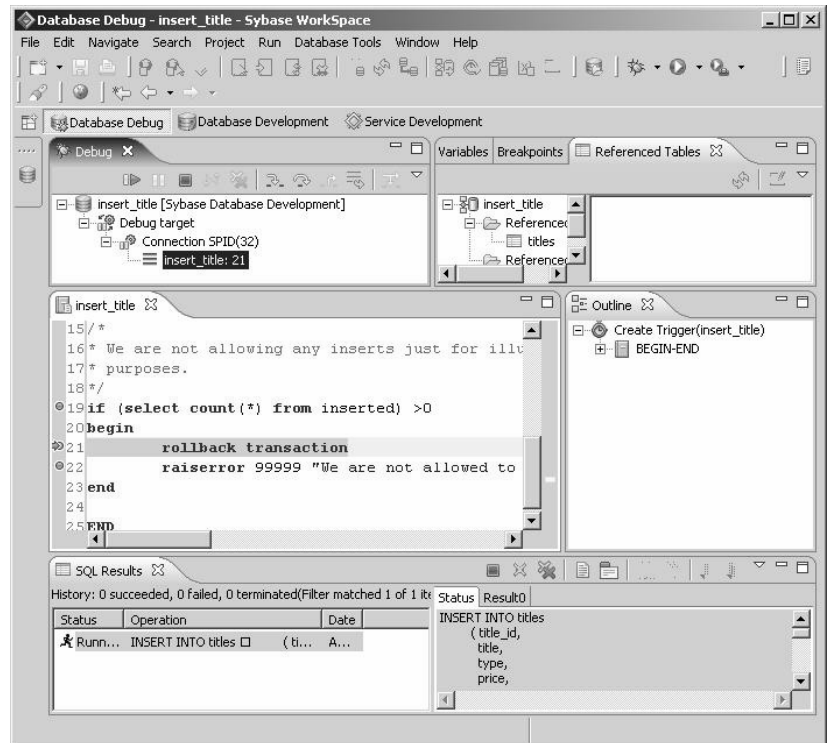
The expanded select from inserted statement executes, and the trigger returns the contents of the inserted special table. You can see the contents of the inserted table in the Result0 tab in the SQL Results view.

- 6 Click the **Resume** icon, located on the **Debug** view toolbar, only once to go to the next breakpoint.



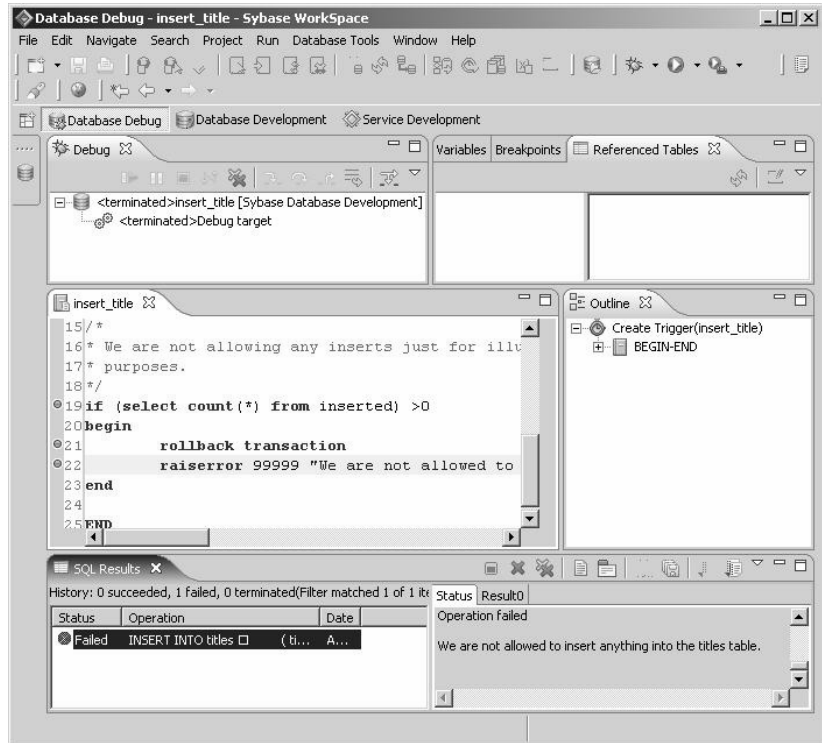


The **if** statement evaluates to true and the trigger enters the begin-end block.



- 7 Click the **Resume** icon two more times to step through the remainder of the trigger.

The initiating INSERT statement fails as expected because the trigger was designed to print an error message during any attempt to insert data in this table (raiserror command).



8 Select **File|Close** on the main menu bar to close the trigger.

You now know how to view an existing trigger, create a new trigger, and run and debug a trigger. You also learned how to query the contents of the *inserted* special table to view its contents during debugging.