SYBASE[®]

Component Tutorials: Web Application Development

Sybase® WorkSpace

1.5

DOCUMENT ID: DC00509-01-0150-01

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 Client/Connect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open Server, Open Solutions, Optima++, Partnerships that Work, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, Pharma Anywhere, Physical Architect, Pocket PowerBuilder, Pocket Builder, 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, 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, Warehouse Architect, 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

About This Book.		v
CHAPTER 1	Getting Started with Web Application Development Web Application Development environment Web Application Development perspective Debug perspective Web Application Development tools Installing the Web Application Development component Configuring the runtime application server Using the Web application project template	1 1 2 3 3 5 6 6
CHAPTER 2	Creating a Web Application Development Project Creating a project Lesson 1: Open Web Application Development perspective Lesson 2: Create a Web Application Development project Lesson 3: Indicate the component-support options	7 7 8 2
CHAPTER 3	Using Java Managed Beans 1 Creating a Java managed bean 1 Lesson 1: Create a Java managed bean 1 Lesson 2: Create a Web page using the bean 2	5 5 3
CHAPTER 4	Using Service Managed Beans 4 Creating a service managed bean on a Web page 4 Lesson 1: Create a service managed bean 4 Lesson 2: Create and test a Web page that uses the bean 4	1 1 9
CHAPTER 5	Linking Web Pages	777

CHAPTER 6	Using DataWindow Objects	63
	Creating a DataWindow object	63
	Lesson 1: Import a DataWindow library	63
	Lesson 2: Create a DataWindow object	72
	Lesson 3: Create Web pages that use DataWindow objects.	77
	Using DataWindow advanced features	88
	Lesson 1: Add server-side events	89
	Lesson 2: Add client-side events	. 97
CHAPTER 7	Debugging Web Applications	101
	Debug perspective	101
	Available debugging tools	106

About This Book

Audience	This guide is intended for users who want to learn how to build servic oriented Web applications using Sybase® WorkSpace integrated development tooling.		
How to use this book	This	s guide contains these chapters:	
	•	Chapter 1, "Getting Started with Web Application Development" introduces Web Application Development, which utilizes the JavaServer Faces (JSF) framework to enable the development, testing, debugging, and deployment of a Web application on a runtime server.	
	•	Chapter 2, "Creating a Web Application Development Project" describes how to use Web Application Development to create a project.	
	•	Chapter 3, "Using Java Managed Beans" shows you how to create a Java managed bean, create a Web page that invokes it, and how to test the Web page by running it on a server.	
	•	Chapter 4, "Using Service Managed Beans" allows you to create a Web page that invokes a SOAP service using a service managed bean.	
	•	Chapter 5, "Linking Web Pages" describes how to link a sequence of Web pages using navigation rules and how to test the linked pages.	
	•	Chapter 6, "Using DataWindow Objects" illustrates how to create a DataWindow® object, how to implement DataWindow objects in a Web page, and how to test the Web page by running it on a server.	
	•	Chapter 7, "Debugging Web Applications" introduces the Debug perspective in Sybase WorkSpace and how to set breakpoints in your Web application code.	

Related documents	For more information on Sybase WorkSpace:
	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 Product Manuals 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 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 <i>SyBooks Installation Guide</i> on the Getting Started CD, or the <i>README.txt</i> 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 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.

The following formatting conventions are used in this manual:

Formatting example	To indicate		
command names and	When used in descriptive text, this font indicates		
method names	keywords such as:		
	Command names used in descriptive text		
	• C++ and Java method or class names used in descriptive text		
	Java package names used in descriptive text		
	Italic font indicates:		
myCounter variable	Program variables		
Server.log	• Parts of input text that must be substituted		
myfile.txt	Directory and file names		
sybase\bin	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.		

Formatting example		To indicate		
	File Save	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.		
		The vertical bar indicates:		
	parse put get	• Options available within code		
	Name Address	• Delimiter within message examples		
		Monospace font indicates:		
	create table	• Information that you enter on a command line or as program text.		
	table created	• Example output fragments		
	Type the Name of the attribute.	GUI field or button name that is the recipient of a procedural action.		
	Click Apply.			
	setup -is:tempdir < <i>full</i> path to alternate temp directory>	Information that must be supplied by the user is displayed between brackets.		
help	Each Sybase installation	n that has purchased a support contract has one or i		

If you need 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.

CHAPTER 1

Getting Started with Web Application Development

Sybase WorkSpace introduces a new set of tooling features for Java Web application development. Web Application Development utilizes the JavaServer Faces (JSF) framework to enable the development, testing, debugging, and deployment of a Web application on a runtime server.

Web Application Development supports the building of JavaServer Pages (JSP) for JSF and HTML applications, enabling both visual and codeoriented development. Additionally, it provides an integrated environment for building Web applications that interface with public SOAP service invocations and for incorporating DataWindow technology.

Web Application Development environment

Web Application Development enables rapid Web application development by providing a structured, integrated development environment.

The following perspectives and tools support both visual and codeoriented development.

Web Application Development perspective

The Web Application Development perspective provides a set of views, wizards, and editors that help you create managed beans and other resources, design Web pages, and configure and test a Web application.



Debug perspective

The Debug perspective provides a set of views and editors that help you debug and troubleshoot a Java Web application.

File Edit Source Navigate Search Project Run Window Hel	p
] E¹ • E] ≙] 棽 • Q • 9a • 9a •] []] @ A']	♥ ♥ \$ • \$ •
😰 🏇 Debug 🖾 Web Application Development	
🖓 Debug 🕴 Servers 🗖 🗖	🕅 🕬= Variables 🙁 Breakpoints 🏦 🖅 🗖 🗖
P Thread [http:2020-Processor 15] (Running) P Thread [http:2020-Processor 15] (Running) P Thread [http:2020-Processor 17] (Running) P Thread [http:2020-Processor 17] (Running) P Thread [http:2020-Processor 17] (Running) P Thread [http:2020-Processor 21] (Running) P Thread [http:2020-Processor 22] (Running) P	 P: ● this=login_jsp (d=285) ⊕ ● jspx_jpage_context: PageContextImpl (d=320) ⊕ ⊕ pageContext: PageContextImpl (d=325) ⊕ ⊕ pageContext: PageContextImpl (d=325) ⊕ ⊕ oute BodyContentImpl (d=330) ⊕ ⊕ _jspx_th_h_panelGrid_0= PanelGridTag (d=334) ⊕ _jspx_eval_h_panelGrid_0= 1
	<u>×</u>
🛞 Web Browser 🔄 login.jsp 🗙 🖓 🗖	E Outline 🕱 🛛 🕞 🗸 🖓
<pre>http://www.sun.com/st/core http://www.sun.com/st/core Login ID #floginID) Password ch:panelGrid columns="2"></pre>	◇ ispidrective.taglib.uri=http://java.sun.com/jsf/html ◇ ispidrective.taglib.uri=http://java.sun.com/jsf/core Ø DOCTVE:HTML I HML Ø HML Ø OCTVE:HTML Ø OCTVE:H
E Console 🖄 Tasks	
Tomcat v5.0 Server _localhost [Apache Tomcat] C:\Sybase\WorkSpa	ce\JDK1.4.2_06\bin\javaw.exe (May 10, 2006 9:49:52 AM)
INFO: Starting Coyote HTTF/1.1 on http-202 May 10, 2006 9:49:58 AM org.apache.jk.comm INFO: JK2: ajp13 listening on /0.0.0.800 May 10, 2006 9:49:58 AM org.apache.jk.serv INFO: Jk running ID=0 time=0/219 configm May 10, 2006 9:49:58 AM org.apache.stalu	o on.ChannelSocket init 9 er.JkMain start ull a.startun.Catalina start
INFO: Server startup in 3190 ms	~
 	>

Web Application Development tools

	Web Application Development adds these features to Sybase WorkSpace.
Automatic code generation	Available for artifacts that invoke public SOAP services in a Web page and for JSP layouts that are based on managed beans.
DataBinding view	Value and method binding to user interface components.

Faces Configuration editor	JSF application configuration files creation and editing:				
	• Graphical editor to facilitate the linking of Web pages				
	• Form-based editor to easily define and edit the parameters of the <i>faces-config.xml</i> file				
	• Source editor to view the code base of the <i>faces-config.xml</i> file				
Form-based resource bundle editor	Resource bundle message file creation and editing.				
Multi-tab properties view	Web page control properties and attributes definition and editing.				
Object creation wizards	Complete set of wizards to facilitate the creation of Web pages and their components.				
Reuse of DataWindow libraries	Importing of existing DataWindow libraries into Sybase WorkSpace.				
Testing and debugging tools	Web page validation throughout the development life cycle.				
Troubleshooting tools	Assistance with troubleshooting design and development problems:				
	• Problems view to identify whether the Java source code in a JSP page has compilation errors.				
	• Error markers on the source view page to identify the cause and solution for an error. Moving your mouse over an error marker displays the problem cause, and double-clicking the error marker displays possible resolutions.				
	• Error Logs view to identify design-time errors.				
Web page development	Web application development support for:				
	DataWindow objects				
	• EJB managed beans				
	• Java managed beans				
	Service managed beans				
Web Page editor	Graphical and code-based design and editing:				
	• Maximize the design canvas and source view using the resize icons in the main toolbar.				
	• Enter and edit text directly on the design canvas.				
	• Resize all JSF HTML controls on the design canvas.				

- Manipulate all objects on the design canvas.
- Undo and redo all commands entered onto the Web Page editor using the Edit menu in the main menu bar.
- Add cascading style sheet (CSS) styles to any object by simply rightclicking the object.
- Drag and drop controls from a palette directly onto the Web Page editor.
- Drag and drop properties and methods from the Data Bindings view onto the Web Page editor to automatically generate a JSP layout and bind business logic to the controls.

```
XML Web file editor Creation and editing of the web.xml file.
```

Installing the Web Application Development component

Install Sybase WorkSpace 1.5 with the Web Application Development component. If your installation of Sybase WorkSpace does not include the Web Application Development component, you must install it. 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**.

Configuring the runtime application server

Web Application Development supports both the EAServer and Apache Tomcat server in the runtime environment. Before you can perform the Web Application Development tutorials, verify that you installed the Web Application Development features of Sybase Developer Edition Servers. See the *Sybase Developer Edition Servers 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 of the Sybase WorkSpace, which includes the Developer Edition Servers, at http://eshop.sybase.com/eshop/. Once you log in to eShop, in the left pane, click **Development & Integration** and then **WorkSpace**.

Using the Web application project template

If you do not have time to complete the entire tutorial, you can quickly create a project using the Sybase Web Application Sample template. You can then view the Java files, JSP pages, and other project artifacts that compose the completed Web Application Development component tutorial.

For instructions on how to create the sample project, see "Lesson 2: Create a Web Application Development project" on page 8.

CHAPTER 2

Creating a Web Application Development Project

This chapter describes how to create a project in Web Application Development. To efficiently develop and manage a Web application, you must first create a Web Application Development project and associate it with a runtime server. In this tutorial, you use the Apache Tomcat 5.0 runtime server.

Creating a project

The following lessons illustrate how to use Sybase WorkSpace to create a Web Application Development project:

Lesson 1: Open Web Application Development perspective Lesson 2: Create a Web Application Development project Lesson 3: Indicate the component-support options

Lesson 1: Open Web Application Development perspective

First, you must launch Sybase WorkSpace and open the Web Application Development perspective.

Opening the Web Application Development perspective

- 1 Select Start|Programs|Sybase|Sybase WorkSpace|Sybase WorkSpace 1.5 to start Sybase WorkSpace.
- 2 Select **Window|Open Perspective|Web Application Development** to open the Web Application Development perspective.



The Web Application Development perspective displays.

Lesson 2: Create a Web Application Development project

Create the Web Application Development project using the blank project template.

- Creating a Web Application Development project
 - 1 Select File|New|Project from the Sybase WorkSpace menu bar.



The New Project wizard displays.

🗧 New Project			×
Select a wizard Create a new Web application project			
Wizards:			
Java Project Java Project Java Project from Existing Ant Buldfil Plugin Project CVS CVS CVS Designer ECipse Modeling Framework Dava Other Pugin Development Sybase New Project From Template Sybase Web Application Project Sybase	e		
			(?)
< 8%	Next >	T Firish	Cancel

2 Expland the **Sybase** folder, select **Web Application Project**, and click **Next**.

Sybase WorkSpace displays the New Web Application Project wizard.

- 🗧 New Web Application Project New Web Application Project Enter a unique project name and edit other fields as necessary. If you are creating a blank project, click Next to add component support. Project name: Tutorial Project location: E:\swadmain_bmeng_vu\calm\dub\swadiruntime-workspace Browse... Module version: 2.3 ٠ Target runtime: Sybase EAServer v5.x ¥ New... Context root: Tutorial Source folder: src Blank Project • Template: $\ensuremath{\square}$ Open the Developing a Web Application cheat sheet after the project is created. < Back Next > Einish Cancel
- 3 In the **Project name** field, enter Tutorial and click **New**.

Note If you want to see the tutorial project files and artifacts without performing the complete tutorial, select **Sybase Web Application Sample** from the **Template** drop-down list and click **Finish**. In the WorkSpace Navigator, you can now view the Java files, JSP pages, and other project artifacts that compose the completed Web Application Development component tutorial.

- **Configuring the server on which to deploy your Web application**
 - 1 In the **New Server Runtime** dialog box, expand the **Apache** folder and select **Apache Tomcat v5.0**.

🗧 New Server Runtin	18			×
New Server Runtime Define a new installed ser	ver runtime er	nvironment		
Runtimes are used at buik	time to compi	ile projects.		
Select the type of runtime	that you wan	t to define:		
		Do	nt see your serve	er listed? Click here
A Apache Tomo A Apac	at v4.0 at v4.1 at v5.0 at v5.5 wer v5.x			
			View By: Ve	ndor 💌
Description: Apache Tomo	et v5.0 suppor	rts J2EE 1.2, 1	.3, and 1.4 Web r	Cancel

2 Click Next.

Sybase WorkSpace displays the **Tomcat Server** page.

🗧 New Server Runtime	×
Tomcat Server Specify the installation directory	R
Name:	
Apache Tomcat v5.0	
Tomcat installation directory:	
E:\Sybase\WorkSpace\DevRuntimes\Tomcat	Browse
JRE:	
jdk1.5.0_06	Installed JREs
< Back Next >. Finish	Cancel

3 In the **Tomcat installation directory** field, click **Browse** to specify the location of the Tomcat server:

<installation directory>\DevRuntimes\Tomcat

4 Click **Finish** to return to the **New Application Project** page.

Lesson 3: Indicate the component-support options

This tutorial demonstrates how to build a Web application using the blank project template. However, if you do not have time to complete the tutorial, you can create a project using the Sybase Web Application Sample template, which creates the Java files, JSP pages, and other project artifacts that compose the completed Web Application Development component tutorial.

- Indicating component-support options
 - 1 In the New Web Application Project wizard, click Next.
 - 2 In the **Add Component Support** page, click all check boxes to select support for all the Web Application Development components and click **Next**.



Туре	Description
Sybase DataWindow Support	Enables the use of DataWindow objects in Web application development.
JavaServerFaces Support	Enables the use of JSFs in Web application development.
SOAP Service Support	Enables the use of SOAP services in Web application development.

Table 2-1: Selecting Web Application Development support

Sybase WorkSpace displays the Project Reference page.

- 3 In the **Project Reference** page, click **Next**.
- 4 Review the **Summary** page.

> New Web Application Pr	oject	×
Summary A resource will be created with th	ne following values. Click Back to change a value.	
Field	Value	
Web Application Project Name Project Path Template	Tutorial C:\MyWorkspace Blank Project	
<		>
	< Back Next > Finish	Cancel

5 Click Finish.

The Tutorial project appears in the Workspace Navigator view. You are now ready to create a Web page with login and password fields using a Java managed bean. See Chapter 3, "Using Java Managed Beans" for step-by-step instructions.

CHAPTER 3 Using Java Managed Beans

This chapter shows you how to create a Java managed bean, create a Web page that invokes it, and how to test the Web page by running it on a server.

The JavaServer Faces (JSF) framework supports the Model-View-Controller (MVC) design paradigm. It allows you to create managed beans, a reusable software component for business objects whose properties are displayed on Web pages, based on Java classes. You can use value binding to bind bean properties and business logic to the user interface components of the JSP page.

Creating a Java managed bean

The following lessons illustrate how to create a Java managed bean that creates login and password fields on a Web page.

Lesson 1: Create a Java managed bean Lesson 2: Create a Web page using the bean

Before you can perform the steps in this tutorial, create a Web Application Development project using the blank project template as described in Chapter 2, "Creating a Web Application Development Project."

Lesson 1: Create a Java managed bean

Create a Java managed bean based on a Java class.

- Creating a Java managed bean
 - 1 In the **Web Application Development perspective**, click the **Data Bindings** view at the bottom of the perspective window to display it.



2 Right-click **session** and select **New** from the context menu to open the **Create a Managed Bean** wizard.



3 In the **Create a Managed Bean** wizard, select **Java Managed Bean** and click **Next**.



4 In the Faces-Config File page, accept the default settings and click Next.

🗧 New Manage	d Bean Wizard	×
Faces-Config Fi Select a faces-con bean.	le fig file, such as faces-config.xml, to save the new managed	Ê
Project name:	Tutorial	•
Faces-Config file:	/Tutorial/webroot/WEB-INF/faces-config.xml	Browse
	< Back Next > Birsh	Cancel

5 In the Java Class Selection page, select Create a new Java class and click Next.



- 6 In the **Java Class** page, define the Java class:
 - In the **Package** field, enter com.sybase.webapp.tutorial.
 - In the Name field, enter LoginBean.

lava Class	dage	
Create a New Java	0855	-
Source folger:	Tutorial/src	Browse
Package:	com.sybase.webapp.tutorial	Browse
Enclosing type:	[]	Brogse
Name:	LoginBean	
Modifiers:	♥ gubic C default C prigate C protected □ abstract □ final □ static	
Superdass:	java.lang.Object	Browse
Interfaces:		Add
		Remove
Which method stub:	would you like to create?	
	Constructors from superclass	
	Ingerited abstract methods	
Do you want to add	comments as configured in the properties of the current proje Generate comments	ac.4

Accept the remaining default values and click **Next**. Sybase WorkSpace displays the Managed Bean Configuration page.

7 In the **Managed Bean Configuration** page, click **Add**.

The Create Property dialog box displays.

Browse

- 8 Enter the property information for the login ID:
 - In the **Property name** field, enter loginID.
 - In the **Property class** field, enter java.lang.String.
 - In the Value field, enter sybase.

The Value type field displays **value**. Keep the rest of the defaults and click **OK**.

- 9 Click Add again to enter the property information for the password.
 - In the **Property name** field, enter password.
 - In the **Property class** field, enter java.lang.String.
 - In the Value field, enter sybase.

The Value type field displays **value**. Keep the rest of the defaults and click **OK**.

10 In the New Managed Bean wizard, click Next.

venerai	IssisBase		
cope: escription:	session		
roperties:		1	
Property loginID password	java.lang.5 java.lang.5	Tring Yes Tring Yes	Edit

11 Review the Summary page.

Field	Value
Project name	Tutorial
Faces-Config file name	/Tutorial/webroot/WEB-INF/faces-config.xml
Source folder	Tutorial/src
Package name	com.sybase.webapp.tutorial
Type name	LoginBean
Managed bean name	loginBean
Managed bean scope	session

12 Click **Finish** to create the Java managed bean.

Lesson 2: Create a Web page using the bean

Create a Web page and bind it to the business logic in the Java managed bean you just created. Then, test the Web page by running it on the runtime server.

* Creating a Web page

1 Select File|New|Web Page from the menu bar.

HIE Edit Navigate	Search Project Run	window Help	
New	Alt+Shift+N	Project	•
Open File		CO Ealdan	
Close	Ctrl+F4	- Folder	
Close All	Ctrl+Shift+F4	File	
🗐 Save	Ctrl+S	G Class	
Save As.		Uf Interface	531-647
Save áll	CHALSHERLS	Web Application P	roject
Revert	CONSIGUES	해양 JSF Configuration	File
Hereix		🖉 Managed Bean	
Move		Resource Bundle F	ile
Rename	F2 F5	📲 Web Page	
Convert Line Delimi	ters To	ES Example	
🖹 Print	Ctrl+P	E [®] Other	Ctrl+N
Switch Workspace.	••)		
🖄 Import			
🛃 Export			
Properties	Alt+Enter		
Exit		-	

The New Web Page File wizard opens.

≫ New Web Page File				×
Select the project in which you want to creat	e the Web page f	ile.		\square
Enter or select the parent folder:				
Tutorial/webroot				
MyMobileServer Tutorial Settings Subjective Generated Src Webroot				
rie nane, ji	< Back	Next >	Finish	Cancel

2 In the **File name** field, enter login.

nter or select the parent folder: Tutorial/webroot		
ile negge: Togin		

3 Click **Finish** to create the *login.jsp* page.

Sybase WorkSpace creates the *login.jsp* page and displays its preliminary contents in an editor.

Generating the page layout

Generate the layout of the *login.jsp* page, using the login bean that you just created.

1 In the **Data Binding** view, expand the tree to see the contents of **loginBean**.



2 Drag and drop **loginBean** to the design pane of the *login.jsp* page to open the **Create JSF Components** wizard to create JSF components that map to the managed bean's attributes.

Create JSF Co	mponents			[
Create JSF comp (i) Navigate to find	conents to map to the attributes that you	D managed bean attr want to use. Select the attr	ibutes. ibutes.	\square^{\diamond}
Page type: Inputt	ing data 💌			
Bean structure	Mapping to JSF comp		Label	_
V loginbean		inputText	Login ID	Move up
	☑ password	inputText	Password	Move dowr
				Select all
				Deselect al
		< Back Next	> Finish	Cancel

3 Accept the default settings for the **loginID** class member.
4 To define the JSF component for the **password** class member, click the **JSF component** column in the password row, select **inputSecret** from the drop-down list, and click **Next**.

Create JSF Co Create JSF comp Navigate to find	mponents ponents to map t the attributes that you	o managed bean attri I want to use. Select the attri	butes. butes.	
Page type: Input	ing data 📃 💌	oonent		
loginBean	Attribute	JSF component	Label	
	 ✓ loginID ✓ password 	inputText inputSecret inputSecret	Login ID	Move up Move down
		inputText		Select all
		outputLink outputText	×	Deselect all
		< Back Next :	> Finish	Cancel

5 On the next page, accept the defaults and click **Finish** to generate a panel grid layout.

reate JSF components	to map to managed bean attributes.	
Configure a container to hold	the generated components.	
Generated container		
🔽 Use a table-typed contai	iner to hold components.	
JSF panelGrid		
C HTML table		
Number of component (pairs for each row: 1	
Codes to be generated		1.00
<h:form> <h:panelgrid <br="" columns="2'
<h:outputText value="><h:outputtext value="</th><th>'>
Fault"></h:outputtext> ⊯</h:panelgrid></h:form>		
<pre>{locallimeSoap.locallimeBy2</pre>	ipCodeInvoker.rault/"> Zip Code">	
value="#	inCadeInvolver input ainCade)">	<u>×</u>
Value="# //ocolTimoCoop_locolTimoPu7		
Value="# (localTimoCopp localTimoDu?		
Value="# //ocstTimeCoop locstTimeDu?		

Adding a messages control

Add a messages control to the Web page to display a message when the user inputs do not pass the validation of the business logic.

- 1 In the Web Page editor, select the **Design** tab, if necessary.
- 2 Place your cursor in the design pane at the end of the **Submit** button, and press **Enter** twice to add two hard breaks in the Web page.

The line breaks appear as

 in the Web Page editor's source pane.



3 Select **Window**|Show View|Palette to display the Palette view.

4 In the **Palette** view, click to expand the **JSF HTML** folder and drag and drop the **Messages** control in the design pane directly below the hard breaks.

🕏 Palette 🗙	- 0
Select	^
III, Marquee	
🕞 HTML	
🗁 JSP	
🗁 JSF Core	
🕞 JSF HTML	*
📺 Column	
CommandButton	
🖉 CommandLink	11
🔲 DataTable	
📋 Form	
🔣 GraphicImage	
InputHidden	
InputText	
InputTextarea	
Message	
Messages	
CutputFormat **	
CutputLabel	
😥 OutputLink	
Se OutputText	
PanelGrid	*

5 Double-click the **Messages** control in the design pane to display the **Quick** Edit tab in the **Properties** view.

	in.java 🚽 *login.jsp 🗙	-
🔮 directive	.page 🗮 http://java.sun.com/jsf/html 🗮 http://java.sun.com/jsf/core	
Login ID	#(loginID)	
Password	4	
Submit		
d		
message		
	•	
	<h:outputtext value="Login ID"></h:outputtext>	^
	<h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#{loginBean.loginID}"><td>></td></h:inputtext>	>
	<h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#{loginBean.loginID}"><h:outputtext value="Password"></h:outputtext> <h:outputtext value="#!loginBean pageneral"></h:outputtext></h:inputtext>	>
<1	<pre><h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#{loginBean.loginID}"></h:inputtext> <h:inputsecret value="#(loginBean.password)"></h:inputsecret></pre>	:> iec
<br <h< td=""><td><pre><h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#{loginBean.loginID}"></h:inputtext> <h:inputsecret value="#(loginBean.password)"> :commandButton value="Submit"></h:inputsecret></pre></td><td>sec</td></h<>	<pre><h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#{loginBean.loginID}"></h:inputtext> <h:inputsecret value="#(loginBean.password)"> :commandButton value="Submit"></h:inputsecret></pre>	sec
<br <h< td=""><td><pre><h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#(loginBean.loginID)"></h:inputtext> <h:inputsecret value="#(loginBean.password)"></h:inputsecret></pre></td><td>> Sec</td></h<>	<pre><h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#(loginBean.loginID)"></h:inputtext> <h:inputsecret value="#(loginBean.password)"></h:inputsecret></pre>	> Sec
<br <h <b< td=""><td><pre><h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#(loginBean.loginID)"></h:inputtext> <h:inputsecret value="#(loginBean.password)"></h:inputsecret></pre></td><td>sec</td></b<></h 	<pre><h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#(loginBean.loginID)"></h:inputtext> <h:inputsecret value="#(loginBean.password)"></h:inputsecret></pre>	sec
<br <h <b< td=""><td><pre><h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#{loginBean.loginID}"></h:inputtext> <h:inputsecret value="#(loginBean.password)"></h:inputsecret><td>sec</td></pre></td></b<></h 	<pre><h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#{loginBean.loginID}"></h:inputtext> <h:inputsecret value="#(loginBean.password)"></h:inputsecret><td>sec</td></pre>	sec
<br <h <b <td><pre><h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#{loginBean.loginID}"></h:inputtext> <h:outputtext value="#(loginBean.password)"><td>sec</td></h:outputtext></pre></td></b </h 	<pre><h:outputtext value="Login ID"></h:outputtext> <h:inputtext value="#{loginBean.loginID}"></h:inputtext> <h:outputtext value="#(loginBean.password)"><td>sec</td></h:outputtext></pre>	sec

6 In the **ID** field, enter Message.

Quick Edit	ID:	Message	8
Attributes	Layout:		
	Global Only:		
	Style:		Edit 😒

7 Next, modify the loginBean class, as described in the next section, "Editing the Java file."

* Editing the Java file

Add the validateLogin() method to the loginBean class.

- 1 In the **Data Bindings** view, right-click **loginBean** and select **Open** to open the *LoginBean.java* file in the editor.
- 2 Add the following import statements under the package statement:

```
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
```

Ignore the warning markers in the marker bar to the left of the Java editor. The warning messages resolve once you add the code for the validateLogin() method.

Note

- If you manually enter the code, the Content Assist tool prompts you with code selection. Double-click the code in the list to select it.
- If you want to undo or redo a code entry, select Undo or Redo from the Edit menu in the menu bar.
- If error markers appear in the marker bar to the left of the editor, hover over the marker to identify the error. It is recommended that you resolve all errors before continuing.
- 3 Add the validateLogin() method into the loginBean class.

```
public String validateLogin()
{
    if (this.loginID.equals("sybase") &&
    (this.password.equals("sybase")))
        return "success";
    FacesMessage fm = new
    FacesMessage(FacesMessage.SEVERITY_ERROR,
        "Invalid Username / Password","");
    FacesContext.getCurrentInstance().addMessage
        ("login button",fm);
    return "failed";
}
```

The formatted code looks like this in the Java editor.

J *Login	iBean. java 🗙 🚽 *login. jsp	- 5
		~
9	<pre>public String validateLogin()</pre>	
	{ if (this.loginID.equals("sybase") && (this.password.equals	
	("sybase"))) return "success".	
	FacesMessage fm = new	
	<pre>FacesMessage(FacesMessage.SEVERITY_ERROR,"Invalid Username / Password","") FacesContext.getCurrentInstance().addMessage("login button",fm); return "failed";</pre>	;
	3	
		~

4 Select **File**|**Save** from the menu bar to save the *LoginBean.java* file.

The Data Bindings view displays the validateLogin() method.

Properties 😤 Data Bindings 🗙	
□ · · · · · · · · · · · · · · · · · · ·	~
o loginID o password	
→ 🔄 application	<u>~</u>

- 5 To associate the validateLogin() method with the **Submit** button, return to the *login.jsp* file.
- 6 Expand the **loginBean** session in the **Data Bindings** view and drag and drop the validateLogin() method onto the **Submit** button in the design pane.

Sybase WorkSpace adds the following code to the JSP page.

```
<h:commandButton value="Submit"
action="#{loginBean.
validateLogin}"></h:commandButton>
```

7 View the following code change in the Web Page editor's source pane.

🖞 LoginBean.java 🛛 🖶 *login.jsp 🗙	-
🕙 directive.page 🗮 http://java.sun.com/jsf/html 🗮 http://java.sun.com/jsf/	core
<> HEAD	
Login ID #(loginID)	
Password *****	
Submit a	
messages	
<pre></pre>	utText>
<h:inputsecret value="#{loginBean.passw</td><td>ord}"></h:inputsecret>	
<h:commandbutton action="#(1</td><td>oginBean.validateLogin</td></tr><tr><td></td><td></td></tr><tr><td><h:messages id=" message"="" value="Submit"></h:commandbutton>	
/ HTWIN	
N/ III III /	

Editing the attributes for the Submit button

- 1 Double-click the **Submit** button in the design pane to display the **Quick Edit** tab in the **Properties** view.
- 2 In the ID field, enter loginbutton.
- 3 In the Value field, overwrite the default value, Submit, and enter Login.

Properties	× Data Bindir	igs	h:commandButton 👻 🗖 🕻
Quick Edit	ID:	loginbutton	9
Attributes	Value:	Login	6
	Action:	#{loginBean.validateLogin}	8
	ActionListener:		Browse
	Type:		▼ @
	Style:		Edit
	Listeners		Action
			Add
			Remove

- 4 Select **File**|**Save** from the menu bar to save the *login.jsp* file.
- 5 Select the **Preview** tab at the bottom of the editor to preview the Web page.

🗋 LoginBean.java 🛛 🚽 login.jsp 🗙	- 8
Login ID #{loginBean.loginID} Password •••••	×

Testing the login.jsp Web page

Run the Web page on the Tomcat 5.0 server.

1 In the WorkSpace Navigator, expand the **webroot** folder, right-click *login.jsp* and select **Run** from the context menu.



Note If you receive an error, compile the Java code. In the **WorkSpace Navigator** view, right-click **Tutorial** and select **Build** from the context menu. 2 In the **Run On Server** wizard, select **Choose an existing server**, and select **Tomcat v.5.0 Server** @ **localhost** from the list.

	×
Define a New Server Choose the type of server to create	
How do you want to select the server? C Choose an existing server C Manually define a new server Select the server that you want to use:	
िि⊸ localhost ि २२ Tomcat v5.0 Server @ localhost ि	
View By: H Description: Apache Tomcat v5.0 supports J2EE 1.2, 1.3, and 1.4 Web modules.	ost name 💌
< Back Next > Finish	Cancel

If you do not see an existing server, you must define one. See "Starting the Tomcat 5.0 server" on page 38 for step-by-step instructions.

3 Click Finish.

The Apache Tomcat server starts, and the JSF Page Template view displays your Web page.

4 To test the Web page in the browser, enter anything in the **Password** field and click **Login**.

The JSF Page Template view displays Invalid Username/Password. Once you create a service managed bean and link its associated JSF page with this one in subsequent lessons in this tutorial, the password functionality works.

🚽 login.jsp 🔗 JSF Page Template 🗙	
 Invalid Username / Password 	
	~

5 Click the **Stop the server** icon in the **Servers** view to stop the server before continuing.

Properties	Data Bindings	해 Servers	S Console	\$00%	R P - D
Server			Status		Stop the server
± 減 To	mcat v5.0 Serve	er @ localhost	: 🚡 Started		Synchronize
1					
					<u> </u>

You are now ready to create a service managed bean that displays the local date and time for a specified zip code. Continue to Chapter 4, "Using Service Managed Beans."

Starting the Tomcat 5.0 server

If you try to run a JSP page and notice that you do not have a previously defined server, you can manually define and start the Tomcat 5.0 server necessary to run and test the JSP file.

- 1 In the **Run On Server** wizard, click **Manually define a new server**.
- 2 In the Apache folder, click Tomcat v5.0 Server and click Next.

	×
Define a New Server Choose the type of server to create	
How do you want to select the server?	
C Choose an existing server	
Manually define a new server	
Server's host name: localhost	
Select the server type:	
Don't see y	your server listed? Click here
Image: System Image: System <td< td=""><td>ew By: Vendor 💌</td></td<>	ew By: Vendor 💌
Server runtime: Apache Tomcat v5.0	▼ Installed Runtimes
Set server as project default (do not ask again)	
< Back Next >	Finish Cancel

3 In the **Available projects** list, select **Tutorial** and click **Add** to add the project to the **Configured projects** list.

			×
Add and Remove Project Modify the projects that are cor	s nfigured on the server		
Move projects to the right to con Available projects:	figure them on the server Add > <remove add="" all="">> <<remove all<="" th=""><th>Configured projects:</th><th></th></remove></remove>	Configured projects:	
	< Back Next >	Finish	Cancel

4 Click **Finish**.



The Servers view displays the server

You are now ready to run the JSP file on the server.

- To run the *login.jsp* page, see "Testing the getTimeService.jsp Web page" on page 53.
- To run the *getTimeService.jsp* page, see "Testing the getTimeService.jsp Web page" on page 53.

CHAPTER 4 Using Service Managed Beans

Web Application Development allows you to create a Web page that invokes a SOAP service using a service managed bean.

This chapter describes how to create a service managed bean, how to create a Web page that invokes the service managed bean, and how to test the Web page by running it on a server.

Creating a service managed bean on a Web page

The following lessons illustrate how to create a service managed bean based on a public WSDL file. In this example, your Web page displays the local date and time for a specified zip code.

Lesson 1: Create a service managed bean Lesson 2: Create and test a Web page that uses the bean

Before you can perform the steps in this tutorial, you must complete the tutorials in all the previous chapters in this guide.

Lesson 1: Create a service managed bean

Create a service managed bean based on a public WSDL file.

Creating a service managed bean

1 If necessary, close the **JSF Page Template** view.

2 Click the **Data Bindings** view at the bottom of the perspective window to display it.



3 Right-click **session** and select **New** from the context menu to open the **Create a Managed Bean** wizard.



4 In the **Create a Managed Bean** wizard, select **Service Managed Bean** and click **Next**.

🗇 Create a Managed Bean 🛛 🔀
Managed Bean Selection Page Creates a service managed bean in the current project.
Select the managed bean type: DataWindow Managed Bean EB Managed Bean Java Managed Bean Service Managed Bean
< Back Next > Finish Cancel

5 In the Faces-Config File page, accept the default settings and click Next.



6 Select Input WSDL and click Next.

🗧 New Managed Bean Wizard	
Service Selection Select a service through inputing it's word file.	
Service Source	
< Back Next >	Frish Cancel

7 To define the WSDL file for the service managed bean, select **From URL** and enter the following in the **URL** field:

```
http://www.ripedev.com/webservices
/LocalTime.asmx?WSDL
```

🗧 New Mana	iged Bean W	izard			×
Service Mar Specify the loc	aged Bean ation of the WS	IDL file for the service.			Ċ
C From Local	Elle				
(€ From LIRL					growse
URL: If the WSDL User <u>n</u> ame: <u>P</u> assword:	http://www.ri document requ	pedev.com/webservices/Lr ires authorization, enter a	ocalTime.asmo user ID and p	c7WSDL Nassword.	
Validate WSD	_				
				R	
		Sack Nex	t>	Brish	Cancel

8 Click Validate WSDL to test the validity of the WSDL file.



- 9 Click **OK** to close the **Validation** box and then **Next** in the **Service Managed Bean** page to continue.
- 10 On the **Service Selection** page, define the service and port for the service managed bean.
 - In the **Service Name** field, if blank, select **LocalTime** from the dropdown list.
 - In the **Port Name** field, select **LocalTimeSoap** from the drop-down list.

Leave the Operation Name	e field blank and click Next.
--------------------------	-------------------------------

♦ New Managed Bean Wiz	zard	X
Service Selection Select the service name, port n	ame and operation name.	
Service Name:		9 10
LocalTime		
Port Name:		
LocalTimeSoap		•
Operation Name:		
		.
If a service requires authorizad	tion, enter a user ID and password.	
	< Back Next >	Finish Cancel

For this tutorial, the the managed bean is based on the port name; however, you can also create service managed beans based on operation names.

11 In the **Managed Bean Configuration** page, accept the default settings and click **Next**.

> New Man	iged Bean Wizard	
fanaged B i Enter a unique description.	ean Configuration name for the managed bean, select the scope, and enter a	
General		
Name:	localTimeSoap	(100)
Scope:	session	
Description:		~
		~
	< Back Next > Finis	t Cancel

12 Review the Summary page, and click **Finish** to create the service managed bean.

New Managed Bean	Wizard
Cummary A new service managed be change a value.	an will be created with the following values. Click Back to
Field	Value
WSDL Location Service name Port name	http://www.ripedev.com/webservices/LocalTime.asmx?WSDL LocalTime LocalTimeSoap
Operation name	lass transformer
Managed bean name	localTimeSoap
٢	
	< Back Next > Finish Cancel

The localTimeSoap service managed bean appears in the Data Bindings view.



Lesson 2: Create and test a Web page that uses the bean

Next, create a Web page, bind it to the business logic in the service managed bean you just created, generate the Web page layout, and then test the Web page by running it on a server.

Creating the Web page

1 Select **File**|**New**|**Web Page** from the main menu.

File	Edit Navigate	Search Project Run	Window Help		
	New	Alt+Shift+N ▶	ĒŜ Project	þ.	
	Open File				
	Close Close All	Ctrl+F4 Ctrl+Shift+F4	Folder		
	Save	Ctrl+S	Class Class		
8	Save As		Web Application D	uniast	
1	Save All	Ctrl+Shift+S	All 1SE Coofiguration	File	
Revert Move	Revert		Managed Bean		
	Move		Handged Bear	=ile	
	Rename	F2	S Web Page		
	Refresh	F5			
	Convert Line Delimit	ters To	Example		
1	Print	Ctrl+P	E [®] Other	Ctrl+N	
	Switch Workspace.				
è	Import				
4	Export				
	Properties	Alt+Enter			
_	Evit				

The New Web Page File wizard opens.

New Web Page File				×
reate a New Web Page File Select the project in which you want to creat	e the Web page I	file.		
Enter or select the parent folder:				
Tutorial/webroot				
WyMobileServer Tutorial Settings build generated src webroot				
≂ile name: ∏				
	< Back	Next >	Finish	Cancel

- 2 In the **Create a New Web Page File** page, make sure the parent folder is *Tutorial**webroot*.
- 3 In the **File name** field, enter getTimeService.
- 4 Click **Finish** to create the *getTimeService.jsp* page.

Sybase WorkSpace creates a *getTimeService.jsp* file with default JSF contents and opens it in the Web Page editor.

Mapping the JSF components to the beans attributes

Map the appropriate JSF components to the service managed bean attributes and then generate the layout of the *getTimeService.jsp* page using the service managed bean that you just created..

 In the Data Bindings view, drag and drop the LocalTimeSoap bean onto the design pane of the *getTimeService.jsp* page to open the Create JSF Components wizard to create JSF components that map to the managed bean's attributes. As required, expand the wizard and the columns to see the entire attribute name and/or label.

2 Move the **Fault** class member to the bottom of the list.

Select the row and click **Move Down** multiple times until the *Fault* class member is at the bottom of the list.

Page type: Inputting data	•			
ean structure	Mapping to JSF component			
- 🔽 localTimeSoap	Attribute	JSF component	Label	
VocalTimeByZipCodeInvo VocalTimeByZipCodeInvo Votat Votat votat	IocalTimeByZipCodeInvoker.in IocalTimeByZipCodeInvoker.o IocalTimeByZipCodeInvoker.st	inputText outputText outputText	LocalTimeByZipCode LocalTimeByZipCodeResult Status	Move up Move dowr
	localTimeByZipCodeInvoker.f	outputText	Fault	Select a
				Deselect
			1	

3 Accept the default values in the **Label** field and click **Next**.

4 Make sure that **JSF panelGrid** is selected as the generated container, accept the remaining default values, and click **Finish**.

Create JSF Components		×
Create JSF components to Configure a container to hold the	map to managed bean attributes. generated components.	
Generated container		
Vise a table-typed container	to hold components.	
JSF panelGrid		
C HTML table		
Number of component pairs	s for each row: 1	
<pre><h:outputtext value="#</pre></th><th>lt"></h:outputtext></pre>	×	
<pre><n:outputtext #<="" <h:outputtext="" pre="" value="Zip' <h:inputText value=" {localtimesoap.localtimebyzipco=""></n:outputtext></pre>	odeInvoker.fault)"> Code">	
∬llocalTimo⊊oan localTimoPu7inCe	ndaTauakar innut ninCada)"s zihuinnutTauts	
	< Back Next > Finish	Cancel

The Create JSF Components wizard generates a Web page grid with ZipCode as the input value and Local Time as the output value.

- 5 Select **File**|**Save** from the main menu bar to save the *getTimeService.jsp* page.
- 6 Select the **Preview** tab at the bottom of the editor to preview the Web page in the **Web Browser** view.

- Testing the getTimeService.jsp Web page
 - 1 In the **WorkSpace Navigator**, expand the **webroot** folder, right-click the *getTimeService.jsp*, and select **Run** from the context menu.



2 In the **Run On Server** wizard, select **Choose an existing server**, and select **Tomcat v.5.0 Server** @ **localhost** from the list.

Note If you completed all previous tutorials in this guide, you already defined the Tomcat server. However, if it does not appear on the list, see "Starting the Tomcat 5.0 server" on page 38 for step-by-step instructions.

🛇 Run On Server
Define a New Server
Choose the type of server to create
How do you want to select the server?
Choose an existing server
C Manually define a new server
Select the server that you want to use:
Tomcat v5.0 Server @ localhost
View By: Host name
< Back Next > Finish Cancel

3 Click Finish.

The Apache Tomcat server starts, and the JSF Page Template view displays your Web page.

- 4 To test the Web page, enter a zip code in the **Zip Code** field, and click **Submit**.
- 5 View the results in the Local Time By Zip Code Result field.

6 Click the **Stop the server** icon in the **Servers** view to stop the server before continuing.

Properties	Data Bindings	해 Servers	Console	\$ 0 p	% N IP
Server			Status		Stop the conus
± ∦ To	mcat v5.0 Servi	er @ localhost	Started		SALCHOUSE

You are now ready to link the two Web pages you previously created, *login.jsp* and *getTimeService.jsp*. Continue to Chapter 5, "Linking Web Pages."

CHAPTER 5 Linking Web Pages

This chapter describes how to link a sequence of Web pages using navigation rules and how to test the linked pages.

Web Application Development allows you to define navigation rules that link a sequence of Web pages together using the JSF Application Configuration editor.

Linking Web pages

In previous tutorials, you created two Web pages: *login.jsp* and *getTimeService.jsp*. Next, define navigation rules that link these two pages in a sequence and then test the link:

Lesson 1: Define navigation rules Lesson 2: Test linked pages

Before you can perform the steps in this tutorial, you must complete the tutorials in all the previous chapters in this guide.

Lesson 1: Define navigation rules

You define navigation rules in the *faces-config.xml* file.

Note Before you begin, verify that you have stopped the Tomcat server.

Defining navigation rules

1 In the **WorkSpace Navigator**, expand the *Tutorial\webroot\WEB-INF* folders, right-click the *faces-config.xml* file and select **Open** from the context menu.

The XML file displays in the JSF Application Configuration editor.

N ⁺ faces-config.x	mi ×	- 0
Introducti	on	^
Introduction		0
	JSF Application Configuration Editor Use the JSF Application Configuration editor to edit faces- configuration. Use the pages in this editor to define and edit page	
~~~~	navigation, managed beans, component, converter, validator, render kit and other element configurations.	2
	Start	-
	Start working with the editor by selecting PageFlow and defining page navigations.	
123	Cheat Sheet	
an	Open the cheat sheet to guide you through editor tasks.	
	<u>Help</u>	
	Open the Help to view topics on working with a faces-configuration descriptor.	on V
Introduction Ove	rview Navigation ManagedBean Component Other Source	

2 To select the Web pages to link, select the **Navigation** tab at the bottom of the editor.

3 Drag and drop the *login.jsp* and *getTimeService.jsp* files from the **WorkSpace Navigator** onto *faces-config.xml* file.



- 4 Select **Window**|Show View|Palette to open the Palette view to link the *login.jsp* page to the *getTimeService.jsp* page.
- 5 Select the **Link** control and drag it onto the editor.
- 6 Click the *login.jsp* page and draw a line to the *getTimeService.jsp* page.



7 Click *getTimeService.jsp* to release the cursor and set the link arrow in place.

- 8 In the **Palette** view, select the **Select** control and then select the link arrow.
- 9 Click the **Properties** tab at the bottom the perspective to define the link properties.

詂 Link		
rom Outcome:		
edirect:	false	•
	龄 <b>Link</b> rom Outcome: edirect:	龄 <b>Link</b> rom Outcome: [ edirect: <mark>false</mark>

- 10 In the **From Outcome** field, enter success.
- 11 Select File|Save from the menu bar to save the *faces-config.xml* file.

# Lesson 2: Test linked pages

## Testing the linked Web pages

Run the Web pages on the Tomcat 5.0 server.

- 1 In the **WorkSpace Navigator**, right-click the *login.jsp* file and select **Run** from the context menu.
- 2 Select Choose an existing server and select Tomcat v5.0Server@ localhost from the list.
- 3 Click Finish.

The Apache Tomcat server starts and the JSF Page Template view opens.

4 To test the Web page sequence in the browser, enter sybase as the password and click **Login**.

The getTimeService.jsp page opens in the browser.

5 Enter a zip code and click **Submit**.

The Local Time By Zip Code Result field displays the local date and time.

6 Click the **Terminate** icon in the **Console** view to stop the server before continuing.


# CHAPTER 6 Using DataWindow Objects

A DataWindow is an object that you use to retrieve, present, and manipulate data from a relational database or other data source. DataWindow objects have knowledge about the data they are retrieving. You can specify display formats, presentation styles, and other data properties so that users can make the most meaningful use of the data.

Web Application Development allows you to create new DataWindow objects and associate them with a Web page.

This chapter describes how to create a DataWindow object, how to implement DataWindow objects in a Web page, and how to test the Web page by running it on a server.

## Creating a DataWindow object

The following lessons illustrate how to use Sybase WorkSpace to import a DataWindow library, create a DataWindow object, and use a DataWindow object on a Web page:

Lesson 1: Import a DataWindow library Lesson 2: Create a DataWindow object Lesson 3: Create Web pages that use DataWindow objects

Before you can perform the steps in this tutorial, you must complete the tutorials in all the previous chapters in this guide.

### Lesson 1: Import a DataWindow library

You can import DataWindow libraries into Sybase WorkSpace and then reuse any DataWindow object in a Web page.

#### Importing a DataWindow library

By default, DataWindow libraries extensions display during the import process.

1 To verify that the DataWindow library extensions are selected to display in the **WorkSpace Navigator**, click the **Menu** icon (down arrow) and select **Navigator Extensions**|**DataWindow Library Extension**.

Web Application Development - faces-config.xm	l - Sybase WorkSpace		- 0>
ile Edit View Navigate Search Project Run Window	Help	10. 10. INS.	4.17
₶・□	)   \$ \$ \$ • \$ • ] \$ \$	이 밖 밖 밖 이라 야 가지 🗌	
😭 🔝 Web Application Development			
	1 19 faces-config.xml		
	Select Working Set		
E Servers	Deselect Working Set		
① 一家 Tutorial	Edit Active Working Set		
	Navigator Extensions 🔹 🕨	✓ DataWindow Library Extension	
-	Sorters	Workbench Java Navigator Extens	lon
	Filters	Dependencies Navigator Extension	
	Views •		
v	<ul> <li>Show File Extensions</li> </ul>		
•	<ul> <li>Extensions Toggle</li> </ul>		
• •	Sorters Toggle	getTimeService	
	Filters loggle	-	
	🕀 Expand All		
	Collapse All		
	Link with Editor		
Filter : By Service Name			
Private			
E SecurityService			
- Conc			
	Introduction Overview Naviga	ation   ManagedBean   Component   Oth	er Source
	Properties Data Bindings	iervers 🛿 🔍 Console 🛛 🏷 🔘 🕯	9 % <b>=</b> 19 ⁻ -
	Server	Status	State
	🛨 🐙 Tomcat v5.0 Server @ k	ocalhost 📲 Stopped	Republish

The check mark to the left of the menu item indicates it is already selected. Do not deselect the menu item by clicking on it. 2 In the **Workspace Navigator**, right-click **Tutorial** and select **Import** from the context menu to import an existing DataWindow library into your project.

¢ ۷	/eb Applica	tion Development - faces-cont	íig. xml - Syba	se WorkSpace				_ D ×
File	Edit View	Navigate Search Project Run V	/indow Help					
] [3	• 🛛 🗅	🌣 🔘   🍕 • ] 🗐   🕭 ∧	₽]@]%∋	$\diamondsuit \bullet \Leftrightarrow \bullet \downarrow$	2 일   많 !			Ŧ
ĒŶ	🛐 Web Applie	cation Development						
		ce Navigator 🗙		ces-config.xml 🔀	<hr/>			
		bileServer						
	🗄 🥃 Serve	ers						
	± 100	New	,					
		Open						
		Open in New Window		_				
		Rename						
		💥 Delete	Delete					
		n Paste	Ctrl+V					
		Copy	Ctrl+C			- (		
		Defrech	EE	login		getT	imeService	
		<ul> <li>Refresh</li> </ul>	13	_				
		💼 Import						
	Service E	Export						
	sge borneo L	Add Bookmark						
	h Eilton P	Close Project						
	- On all	Open Project						
	Priva	Build						
	Publi	Deploy						
		🔘 Run		ction Overview	Navigation Ma	nagedBean Cor	popent Other So	urce
		🕸 Debug			( init			
		Ameliate		es Data Bindings	්ම් Servers ∂	Console	\$ 0 P. 2	
		Rup YDoclet				Status		State
		Update WorkSpace Build Path En/	tries	TUINCAL VOLU DERV	er (@ IOCali IOSC	E prohhen		Republish
		Run As	,					
		Debug As	,					
		Profile As	•	C				(10)
	<u> </u>	Team	•					2
	Tutorial	Compare With	,					
		Restore from Local History						
		PDE Toole						
		Properties	Alt+Enter					
			HICTORIC					

The Import wizard displays.

3 Select DataWindow Library (.pbl) and click Next.

Imports a DataWindow Library (or DataWindow file) into the project Select an import source:  Archive file Checkout Projects from CVS DataWindow Library (.pb) ERwin model Existing Projects into Workspace	<u>'</u>
Select an import source: Archive file Checkout Projects from CVS DataWindow Library (.pb) Existing Projects into Workspace Existing Projects into Workspace	
Select an import source: Archive file Checkout Projects from CV5 DataWindow DataWindow Library (.pbl) Rwin model ERwin model Scheme Electrone	2
Checkout Projects from CV5 Checkout Projects from CV5 DataWindow DataWindow Library (.pb) Existing Projects into Workspace Storma Eacharce	2
Checkout Projects from CV5  DataWindow  DataWindow  DataWindow  Existing Projects into Workspace  Storm Restructs	
DataWindow     DataWindow     Library (.pb)     Existing Projects into Workspace     Storeng Eachurge	
Catal Window Library (.pbl) Catal C	
Existing Projects into Workspace     Stored Each war	
Existing Projects into Workspace	
External Features	
External Plug-ins and Fragments	
File system	
E Log File	
B pupeling the	
i ug Proning nie →δ pu-site = site =	
⇒ Proming nicer	
Security Certificate	
SIMILI 8 file	
10 Symptom Database File	~

- 4 In the **From directory** field, click **Browse** and select this path: <*installation directory*>*\sybase_workspace* *web_development**eclipse**plugins* *com.sybase.stf.jmt.template_1.5.0**tutorial* *webroot**WEB-INF**pb*
- 5 Click the **test.pbl** check box in the right pane.
- 6 Verify that the **Into folder** field reads:

Tutorial\webroot\WEB-INF\pb

7 Accept the remaining default settings, and click **Finish**.

🗇 Import DataWindow Library	×
Import DataWindow Library Import DataWindow resources from the local file system	
From directory: C:\Sybase\WorkSpace\sybase_workspace\web_dev v	Browse
Select All Deselect All Into folder: Tutorial/webroot/WEB-INF/pb	Browse
Options  Overwrite existing resources without warning  Create complete folder structure  Create selected folders only	
< Back Next > Finish	Cancel

8 Verify that you successfully imported the *test.pbl* file by expanding its contents in the **WorkSpace Navigator**.



### Importing the connection profile and viewing an imported DataWindow

A connection profile enables connection to servers, message transports, and databases. Before you can open and view a DataWindow, you must connect to the database so you have access to the DataWindow object data sources.

1 Double-click the *startdemo.bat* file to start the Adaptive Server Anywhere server and the sample database.

<installation directory>\DevRuntimes\ASA\startdemo.bat

Adaptive Server Anywhere successfully starts when you see its icon in the Windows taskbar.



- 2 In the **WorkSpace Navigator**, right-click the **d_emp** DataWindow library and select **Open** from the context menu to create a connection profile for the tutorial database that contains the datasources for this DataWindow object.
- 3 In the **Database Connection Profile** dialog box, click **New Connection Profile**.

Database Connection Profile	×
Database Connection Profile	t a profile (and database,
if applicable).	Connect
	OK Cancel

4 In the New Connection Profile wizard, select Sybase ASA and click Next.

♦ New Connection Profile	×
Wizard Selection Page Create a Sybase ASA JDBC connection profile	Ê
Please select the connection profile type: JDBC Sybase ASA Sybase ASE Sybase ASIQ	
< Back, Next > Finish	Cancel

5 In the Name field, enter webapp_db and click Next.

♦ New ASA Conner	ction Profile	$\mathbf{X}$
Create connection Please enter detailed	n profile information	
Name: Description(optional):	webapp_db	
☐ Auto-connect	when the wizard is finished or when	Enterprise Explorer opens.
	< Back Next >	Finish Cancel

6 In the **Specify a Driver and Connection Details** page, click **Test connection** to verify that you can successfully connect to the server.

New ASA Co	onnection Profile	×
Specify a Driv Select a driver fi connection.	ver and Connection Details rom the drop-down and provide login details for the	$\square^{\diamond}$
Connection Fill	ters   Other Properties	
Subace ASA De	for the drop-down:	
Host:	localhost	
Port:	2638	
Database name		
User name:	dba	
Password:	***	
		Test connection
	< Back Next > Finish	Cancel

Do not proceed until the ping succeeds. If the ping does not succeed, see your system administrator for assistance.

In the **Success** dialog box, click **OK** to continue.

- 7 Click **Finish** to create the connection profile.
- 8 In the **Database Connection Profile** dialog box, click **webapp_db** and then **Connect**.

	×
Database Connection Profile           Image: The profile ("webapp_db") associated with this resource profile (and database, if applicable).	is not valid. Select a
ISFDB           Image: MyDatabase           Image: MyMobileMsalesDB           Image: MyMobilePortalDB           Image: MyMobileSampleDB	Connect
	OK Cancel

9 Click **OK** to connect to the *asaDemo* database.

ID	First Name	Last Name	!	Start De	General	Point	er   Print Spi	ec ◀
Header †					Units			
emp_id	emp_fname	emp_Iname		start_c	Normali	zed (0)		
)etail†					Timer In	terval:		
ummary †					0 Calan			7.
Previous	Next	Update	Sort (	Data D∈		(y		-
ooter †			_		T Hide	Gray Li	ine on page l	oreak
:)(00)				>	Web	DataW	/indow	
Employee ID	First Name	Last Name	1	Start De				
102	Erro	6.0.5		00/20				
102	Fran	Whitney		00720				
102	Fran Matthew	Cobb		00/20				
102 105 129	Pran Matthew Philip	Whitney Cobb Chin		01/01				
102 105 129 Previous	Pran Matthew Philip Next	Cobb Chin Update	Sort [	00/20 01/01 02/03 Data De	I			
102 105 129 Previous	Philip Philip Next	Cobb Chin Update	Sort (	00720 01/01 02/03 Data De	Туре	Prompt	Initial Value	Val
102 105 129 Previous	Philip Philip Next	Update	Sort [	00720 01/01 02/03 Data De Name emp_id	Type	Prompt	Initial Value	Val
102 105 129 Previous	Philip Philip Next	Update	Sort [	00720 01/01 02/03 Data De Name emp_id emp_fname	Type long char(20)	Prompt	Initial Value	Val
102 105 129 Previous ?xml version=", d_emp_row ⊖ ⊕ emp_id emp_ieme_ieme_row	Philip Philip Next	Update	Sort [	00720 01/01 02/03 Data De Name emp_id emp_fname emp_lname	Type long char(20) char(20)	Prompt	Initial Value	Val
102 105 129 Previous ?xml version=" d_emp O d_emp_row ⊖ 0 emp_fn □ 0 emp_fn	Philip Philip Next I 1.0" encoding="UT , pp_id iame	Update	Sort [	00720 01/01 02/03 Data De Name emp_id emp_fname emp_Iname start_date	Type long char(20) char(20) date	Prompt	Initial Value	Val
102 105 129 Previous ?xml version=" d_emp_row ∂_emp_row ⊕_@ emp_fn □	Philip Philip Next I 1.0" encoding="UT , pp_id ame ap_fname	Update	Sort [	00720 01/01 02/03 Data De Mame emp_id emp_fname emp_lname start_date birth date	Type long char(20) char(20) date date	Prompt	Initial Value	Val
102 105 129 Previous	Philip Philip Next I 1.0" encoding="UT , ap_id ame ame ame ame	Update	5 Sort [	Name emp_id emp_iname start_date birth_date	Type long char(20) char(20) date date decimal(3)	Prompt	Initial Value	Val

The *d_emp* DataWindow opens in the editor.

### Lesson 2: Create a DataWindow object

Create a DataWindow object to add it to the DataWindow library.

#### Creating a new DataWindow object

1 Verify that the Adaptive Server Anywhere database is running.

Look for the ASA icon in the Windows taskbar.



If you need to start Adaptive Server Anywhere, see "Importing the connection profile and viewing an imported DataWindow" on page 68.

- 2 Select File|New|Other from the menu bar to open the New DataWindow Object wizard.
- 3 Scroll down and expand the **Sybase** folder, select **DataWindow**, and click **Next**.

				×
Select a wizard Create a new DataWindow				
Wizards:				
Sybase DataWindow DataWindow Deployment Profile Message Definition Monitor Design New Project From T Query Sybase WorkSpace Type Mapping Web Application Pro Database Developm Deptabase	/ emplate Project oject nent			
And provide the second se	t			
Services				۲ ۲
	< Back	Next >	Finish	Cancel

The New DataWindow Object dialog box displays.

♦ New DataWind	ow Object				×
Create DataWind Enter a library path	low Objec	t			\$
DataWindow library: DataWindow name:					Browse
Presentation style:	Free Grap Grad Grid Labe	form h Ip			×
		< Back	Next >	Finish	Cancel

- 4 In the **DataWindow** library field, enter *Tutorial*\webroot\WEB-INF\pb\test.pbl.
- 5 In the **DataWindow** name field, enter mycustomer.
- 6 From the **Presentation style** list, select **Freeform** and click **Next**.
- 7 In the **Database Connection Profile** page, select **webapp_db** and click **Next**.

8 In the **Choose DataSource for Freeform DataWindow** page, select **Quick Select** as the data source and click **Next**.

Choose Data Source fo	or Freeform Da	taWindow			?×
	Which gata source	te would you like SQL Select	e to use? Query	External	
	Eetrieve on F		ack Ne	ext > C	ancel

9 In the **Quick Select** page, select **customer** from the **Tables** list, click **Add All** to add all columns to the form, and then click **OK**.

Quick Select 1. Click on table(s) to select or 2. Select one or more columns, 3. (Optional) Enter sorting and selection criteria below.	deselect To tab the	dsplay comments for a le or column, click right mouse button.		OK Cancel
Tables:	Col	umns:		Add All
Comments:	id fna ado oty sta zip pho <b>b</b>	me ne tress tre tr	×	Help
Column: Id	Fname	Lname	-	Addres: ^
Sort: Criteria:				
Or:			Ŷ	~
<				>

10 In the **Select Color and Border Settings** wizard, accept the default settings for the color and border settings, and click **Next**.



11 Review the **Ready to Create Freeform DataWindow** page and click **Finish** to create the DataWindow object.

eader [ Id: id Fname: fname Lname: Iname Address: address				Gene Units Nor	eral   f s malizer	Pointer P.	•
Fname: fname Lname: Iname Address: address				Unit: Nor	s malize	1 ( = )	
Lname: Iname Address: address				1		d (U) b	-
Address: address				Time	r Inter	rval:	-
				Colo	r		
City: city					Wine	dow Backgr ay Line on	, 💌
			>	Γ,	Neb D	ataWindow	1
Id: 101			<b>_</b>				
Fname: Michaels							
Lname: Devlin							
Address: 114 Pioneer Avenue							
City: Kingston			]				
State: NJ							
ml version="1.0" encoding="UTF-16LE" standalone		Name	Type	F	Promot	Initial Valu	IR I
ycustomer	1	id	lona	-	<b></b>		-
mycustomer_row	2	fname	char(	15)	_		-1
	3	Iname	char(	20)	-	-	-
The second secon	4	address	char(	35)	-		-
in fname	-	address altri	char(	203	-		-
Iname	-	city	unar(	162	-		_
	Ь	state	char(	10)	1		_
of anness	<	100					>

12 View the mycustomer (test.pbl) DataWindow in the editor.

### Lesson 3: Create Web pages that use DataWindow objects

In this lesson, you return to the  $d_master$  and  $d_detail$  DataWindow libraries that you imported earlier and create a Web page using DataWindow objects in these libraries.

- Creating a Web page using a DataWindow object
  - 1 Select File New Web Page from the menu bar to open the New Web Page wizard.

File Edit Navigate	Search Project Run	Window Help	
New	Alt+Shift+N	Project	ł.
Open File		CO Falder	
Close	Ctrl+F4	- Folder	-
Close All	Ctrl+Shift+F4	File	
🗐 Save	Ctrl+5	G Class	
Save As	1.11.11.14.1.1	U Interface	1. A. T. A.
Save All	CHILSHELS	Web Application P	roject
Revert	Contrainers	해 JSF Configuration	File
		🖉 Managed Bean	1
Peoame	E2	Resource Bundle F	ile
Refresh	F5	🚽 Web Page	
Convert Line Delin	iters To	Example	
🖨 Print	Ctrl+P	E [®] Other	Ctrl+N
Switch Workspace	•••:		
🖄 Import			
🛃 Export			
Properties	Alt+Enter		
Exit		-	

2 In the **Create a New Web Page File** page, make sure that parent folder is *Tutorial**webroot*.

3 In the File name field, enter department, and click Finish.

♦ New Web Page File				×
Create a New Web Page File Select the project in which you want to create	e the Web page file			
Enter or select the parent folder: Tutorial/webroot				
MyMobileServer MyMobileServer Servers Tutorial Dominication Servers Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication Dominication				
File name: department				
	< Back	Next >	Finish	Cancel

Sybase WorkSpace creates *department.jsp* with the default JSF contents and opens it in a Web Page editor.

4 To open a DataWindow object, in WorkSpace Navigator, expand the Tutorial\webroot\WEB-INF\pb\text.pbl folder, and drag and drop the d_master DataWindow onto the design pane of the *department.jsp* page.



The Web Page editor displays code for that DataWindow and its graphical representation on the design pane.

mycustomer (	(test 🔄 *department.	jsp 🗙 "2	- 0
E http://syba E http://java. <> HEAD	se.com/jsf 💌 directive.page 🗏 sun.com/jsf/core	ttp://java.sun.com/jsf/html 🕄	
Department ID 1000	Department Name dept_name	Manager J ID 1000 💌	
<tit </tit 	LE>JSF Page Templat	e	<u>^</u>
<body> <f:view> <h:f< td=""><td>orm&gt;</td><td></td><td></td></h:f<></f:view></body>	orm>		
	<syb:datawindow connectString=" dataWindowObjec id="d_master"&gt;&lt;</syb:datawindow 	dba/sql/com.sybase.j t="d_master" library /syb:dataWindow>	dbc2.jd List="/
<td>form&gt;</td> <td></td> <td>&gt;</td>	form>		>
Design Preview			

- 5 Select the **Preview** tab in the Web Page editor to preview the Web page.
- 6 Select **File**|**Save** from the menu bar to save the Web page.
- 7 To test the Web page on the Tomcat 5.0 server, in the **WorkSpace Navigator**, right-click *department.jsp* and select **Run** from the context menu.

8 Select Choose an existing server, select Tomcat v5.0Server @ localhost, and click Finish.

🗇 Run On Server	×
Define a New Server Choose the type of server to create	
How do you want to select the server? Choose an existing server Manually define a new server Select the server that you want to use:	
E ⊕ localhost L 24 Tomcat v5.0 Server @ localhost	
View By: Description: Apache Tomcat v5.0 supports J2EE 1.2, 1.3, and 1.4 Web modules.	Host name 💌
< Back Next > Finish	Cancel

Departme ID	nt Department Name	Mana <u>c</u> ID	let	
100	R&D	501	~	
200	Sales	902	*	
300	Finance	1293	~	
400	Marketing	1576	*	
500	Shipping	703	~	

The Apache Tomcat server starts, and the JSF Page Template opens.

9 In the Console view, click the **Terminate** icon to stop the server before continuing.

#### * Adding a detail DataWindow object to a Web page

Add a detail DataWindow object to the Web page.

- 1 At the bottom of the *department.jsp* page, click the **Design** tab to switch to the Web Page editor.
- 2 In the design pane, place your cursor at the end of the DataWindow object and press **Enter** twice to add two line break tags.

These appear as <br > <br> in the Web Page editor source view.



3 In WorkSpace Navigator, expand the Tutorial\webroot\WEB-INF\pb\text.pbl folder and drag and drop the the d_detail DataWindow under the second line break on the design pane of the *department.jsp* page.

3 P23	E *department.js	φ X		
Select ∩ Щ Marquee	≕, http://sybase. ⇔	con√jsf • directive.page≡	, hitp://java.sun.com/jsl	/html≡_http://java.sun.c
JSP     JSF Core     JSF HTML     Sybase	Department d ID 1000 c	Department Name	Manager d ID 1000 💌	a
	Employee Ma ID 1000 10	nager∂ First Name ID DI▼ emp_fname	"Last Name D emp_Iname [	ID ID ICOO
	< <body></body>	Ш		×
	<fiview> <hifo< td=""><td><pre>im&gt; connectString dataWindow(b) id="d_master": br /&gt; br /&gt; connectString connectString</pre></td><td>"dba/sql/com.s; ect="d_master" "dba/sql/com.s;</td><td>ybase.jdbc2.jd libraryList="/ pw&gt; ybase.jdbc2.jd</td></hifo<></fiview>	<pre>im&gt; connectString dataWindow(b) id="d_master": br /&gt; br /&gt; connectString connectString</pre>	"dba/sql/com.s; ect="d_master" "dba/sql/com.s;	ybase.jdbc2.jd libraryList="/ pw> ybase.jdbc2.jd
8	K Design Preview	dataWindowObje id=#d darail#	ect="d_detail"	libraryList="/

4 On the *department.jsp* page, right-click the **Department ID** DataWindow and select **Add Object Link** from the context menu.



The Create objectlink dialog box displays.

- 5 In the Link Name field, enter dept id.
- 6 In the Link URL field, enter /department.jsp.
- 7 In the Link Target field, enter _self.
- 8 In the **Link Arguments** list, click **Add** to define the argument properties and then click **OK**.
  - In the Name field, enter dept.
  - In the Type field, select DW Column from the drop-down menu.

• In the Value field, enter dept_id.

Link Name:	dept_id		•
Link URL:	/department.jsp		Browse
Link Target:	_self		•
nk Arguments	5: 1	1	
nk Arguments Name	s: Type	Value	Add
nk Arguments Name dept	:: Type DW Column	Value dept_id	<u>A</u> dd
nk Argument: Name dept	Type DW Column	Value dept_id	Add Remove
nk Argument: Name dept	Type DW Column	Value dept_id	Add Remove

- 9 Select **File**|**Save** from the menu bar to save the Web page.
- 10 To test the Web page on the server, right-click **department.jsp** and select **Run** from the context menu.

The *department.jsp* page opens in the JSF Page Template.

11 Click any link under **Department ID**.

The bottom of the JSF Page Template window displays the employee details below.

d_emp (test	t.pbl)	🚽 department.js	ip 🕐 Je	iF Page T	emplate	× "2		
$\Rightarrow \Rightarrow \blacksquare$	🖇 http:/	//localhost:2020/T	utorial/departi	ment.fac	es?dept=1	.00 💌	▶	3
Departmen ID	nt Depart	ment Name		Manag ID	jer			~
<u>100</u>	R&D			501	~			
<u>200</u>	Sales			902	~			
<u>300</u>	Finance			1293	~			
<u>400</u>	Marketir	ng		1576	*			
<u>500</u>	Shipping	,		703	~			
Employee ID	Manager ID	First Name	Last I	Name	Depa	rtment D		
102	501 🛩	Fran	Whitney		100	*		1
105	501 🗸	Matthew	Cobb		100	~		
160	501 🗸	Robert	Breault		100	*		
243	501 🛩	Natasha	Shishov		100	~		
247	501 🛩	Kurt	Driscoll		100	~		
249	501 🗸	Rodrigo	Guevara		100	~		
266	501 🛩	Ram	Gowda		100	*		
278	501 🗸	l erry	Melkisetian		100	*		
316	501 🛩	Lynn	Pastor		100	*		~
8 J.				ll.			>	

## Using DataWindow advanced features

DataWindow advanced features support both the creation of server-side and client-side events. You implement server-side events in Java code and client-side events using JavaScript.

This tutorial contains these lessons:

Lesson 1: Add server-side events Lesson 2: Add client-side events

### Lesson 1: Add server-side events

Add server-side events, which are executed during runtime, to your DataWindow objects. DataWindow server-side events are implemented in Java code with a defined event handler interface.

#### * Creating the Web page and adding the DataWindow object

In this tutorial, you will create a Web page called *employee.jsp* to which you will then add server-side events to a DataWindow object. The server-side events enable you to page through the employee information retrieved from the server using the **Previous** and **Next** buttons.

- 1 Select File|New|Web Page from the menu bar to open the New Web Page File wizard.
- 2 In the **Create a New Web Page File** page, make sure that parent folder is *Tutorial**webroot*.
- 3 In the File name field, enter employee, and click Finish.

Sybase WorkSpace creates the *employee.jsp* with the default JSF contents and opens it in a Web Page editor.

4 To add server-side events to the Web page, in **WorkSpace Navigator**, expand the **Tutorial\webroot\WEB-INF\pb\test.pbl** folder and drag and drop the **d_emp** DataWindow onto the design pane of the *employee.jsp* page.



5 To display the properties for the Employee ID DataWindow object, click the object in the design pane and then click the **Properties** view.

The second second second					
🖹 http://syb	ase.com/jsf 🕥 directive	.page 🚉 http://jav	/a.sun.com/jsf/h	itml 🛋 http://	java.:
Employe ID	First Name	Last Name	Start Date	e Birth D	) <b>ate</b>
Previous	Next Up	date Sor	t Data Des	cending	,
					~
CRODVS					
<f:viet< td=""><td>1&gt;</td><td></td><td></td><td></td><td></td></f:viet<>	1>				
<h:< td=""><td>form&gt;</td><td></td><td></td><td></td><td>100</td></h:<>	form>				100
1000	TOTHY				
	<syb:datawindo< th=""><th>W</th><th></th><th></th><th></th></syb:datawindo<>	W			
	<pre>connectStr</pre>	w ing="dba/sql	/com.sybas	se.jdbc2.;	jd
	<syb:datawindo connectStr dataWindow</syb:datawindo 	w ing="dba/sql Object="d_en	/com.sybas p" lib <u>rary</u>	se.jdbc2.j yList="/W]	jd EB
	<pre><syb:datawindo< th=""><th>w ing="dba/sql Object="d_em &gt;<th>/com.sybas p" library Jindow&gt;</th><th>se.jdbc2.j yList="/W]</th><th>jd EB</th></th></syb:datawindo<></pre>	w ing="dba/sql Object="d_em > <th>/com.sybas p" library Jindow&gt;</th> <th>se.jdbc2.j yList="/W]</th> <th>jd EB</th>	/com.sybas p" library Jindow>	se.jdbc2.j yList="/W]	jd EB
1</th <th><pre><syb:datawindo< th=""><th>w ing="dba/sql Object="d_en &gt;<th>/com.sybas pr library Jindow&gt;</th><th>se.jdbc2.j yList="/W]</th><th>jd EB</th></th></syb:datawindo<></pre></th>	<pre><syb:datawindo< th=""><th>w ing="dba/sql Object="d_en &gt;<th>/com.sybas pr library Jindow&gt;</th><th>se.jdbc2.j yList="/W]</th><th>jd EB</th></th></syb:datawindo<></pre>	w ing="dba/sql Object="d_en > <th>/com.sybas pr library Jindow&gt;</th> <th>se.jdbc2.j yList="/W]</th> <th>jd EB</th>	/com.sybas pr library Jindow>	se.jdbc2.j yList="/W]	jd EB
	<pre><syb:datawindow connectStr dataWindow id="d_emp" h:form&gt;</syb:datawindow </pre>	w ing="dba/sql Object="d_em > <td>/com.sybas p" library findow&gt;</td> <td>se.jdbc2.j yList="/W]</td> <td>jd EB ≽</td>	/com.sybas p" library findow>	se.jdbc2.j yList="/W]	jd EB ≽
Sesign	<pre><syb:datawindow connectStr dataWindow id="d_emp" i:form&gt;</syb:datawindow </pre>	w ing="dba/sql Object="d_em > <td>/com.sybas np" library /indow&gt;</td> <td>se.jdbc2.j yList="/Wi</td> <td>jd ER &gt;</td>	/com.sybas np" library /indow>	se.jdbc2.j yList="/Wi	jd ER >
Sesign Previe	<pre><syb:datawindo connectStr dataWindow id="d_emp" :form&gt; w Data Bindings Se Cota Bindings Se</syb:datawindo </pre>	w ing="dba/sql Object="d_en > <td>/com.sybas p" library Jindow&gt; syb:data</td> <td>se.jdbc2. yList="/₩] aWindow ▼ ♡</td> <td></td>	/com.sybas p" library Jindow> syb:data	se.jdbc2. yList="/₩] aWindow ▼ ♡	
1<br esign Previe Properties Quick Edit	<pre><syb:datawindo connectStr dataWindow id="d_emp" ::form&gt; w Data Bindings 56 * ID:</syb:datawindo </pre>	w ing="dba/sql Object="d_em >ervers d_emp	/com.sybas p" library Jindow> syb:data	se.jdbc2.j yList="/Wi wWindow ▼ ♡	
<pre></pre> Properties Quick Edit Attributes	<pre><syb:datawindo< td=""><td>w ing="dba/sql Object="d_em &gt;ervers d_emp [WEB-INF/pb/tes</td><td>/com.sybas p" library findow&gt; syb:data t.pbl</td><td>se.jdbc2., yList="/₩] wWindow ▼ ♡ Browse</td><td>ja = ∑E ≫ ≫</td></syb:datawindo<></pre>	w ing="dba/sql Object="d_em >ervers d_emp [WEB-INF/pb/tes	/com.sybas p" library findow> syb:data t.pbl	se.jdbc2., yList="/₩] wWindow ▼ ♡ Browse	ja = ∑E ≫ ≫
Attributes	<pre><syb:datawindow connectStr dataWindow id="d_emp" :form&gt;</syb:datawindow </pre>	w ing="dba/sql Object="d_em >ervers d_emp d_emp d_emp d_emp	/com.sybas p/ library findow> syb:data t.pbl	se.jdbc2 yList="/₩] aWindow • ♡ Browse	
Operation of the second secon	<pre><syb:datawindo connectStr dataWindow id="d_emp" ::form&gt; W X Data Bindings Se * ID: * Library List: * DataWindow: JNDI Data Source:</syb:datawindo </pre>	ing="dba/sql Object="d_en >ervers d_emp /WEB-INF/pb/tes d_emp	/com.sybas np" library findow> syb:data t.pbl	se.jdbc2. yList="/W] aWindow ★ ♡ Browse ¥	)a ∑E ≫ ≫
Design Previes Properties Quick Edit Attributes	<pre><syb:datawindo connectStr dataWindow id="d_emp" :form&gt; w % Data Bindings Se * ID: * Library List: * DataWindow: JNDI Data Source: OnLoadHandler:</syb:datawindo </pre>	w ing="dba/sql Object="d_en >ervers d_emp /WEB-INF/pb/tes d_emp	/com.sybas np" library findow> syb:data t.pbl	se.jdbc2. yList="/W] Window * ^{\scale} Browse T Browse	ja SB ≫ ≫ % %
A state of the	<pre><syb:datawindo connectStr dataWindow id="d_emp" h:form&gt; W &amp; Data Bindings Se * ID: * Library List: * DataWindow: JNDI Data Source: OnLoadHandler: Rows Per Page:</syb:datawindo </pre>	w ing="dba/sql Object="d_em >ervers d_emp [/WEB-INF/pb/tes d_emp	/com.sybas mp" library Jindow> syb:data t.pbl	se.jdbc2., yList="/W Window • Browse Browse	ja ∑B >> 20 0 0 0 0 0 0
Design Previe Properties Quick Edit Attributes	<pre><syb:datawindo connectStr dataWindow id="d_emp" :form&gt; % % % Data Bindings Se * ID: * Library List: * DataWindow: JNDI Data Source: OnLoadHandler: Rows Per Page: Show Navigation Bat</syb:datawindo </pre>	w ing="dba/sql Object="d_em >ervers d_emp [	/com.sybas p/ library findow> syb:data t.pbl	se.jdbc2 yList="/W] aWindow • Browse Browse	ja EB ≫ ≫ 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

6 In the **Properties** view, select the **Attributes** tab.

7 Locate the **rowsPerPage** attribute under the General category, and enter 10 in the **Value** column to display 10 records per Web page.

Properties	× Data Bindings Servers	syb:dataWindow 👻 🗢	- 0
Ouide Edit	Property	Value	^
QUICKEDIC	itemFocusChanged		
Attributes	onClicked		
	rowFocusChanged		
	rowFocusChanging		=
	🖃 General		
	dataWindowObject	d_emp	
	id	d_emp	
	libraryList	/WEB-INF/pb/test.pbl	
	pageNavigationBarV	í	
	rowsPerPage	10 👻 😨	
			~
			100

- Adding a messages control to the Web page
  - 1 Select **Window**|**Show View**|**Palette** from the main menu to add a messages control to the Web page.

2 In the Palette, expand the **JSF HTML** folder, and then drag and drop a messages control directly after the Employee ID DataWindow on the *employee.jsp* page.

🔄 *employee.jsp	×			- 8
🛋 http://sybase	.com/jsf 🖭 directiv	/e.page 🛋 http://ja	va.sun.com/jsf/html	🛋 http://java.sur
<> HEAD				
Employee / ID	First Name	Last Name	Start Date	Birth Date
1000	emp_fname	emp_Iname	00/00/0000	00/00/0000
Previous	Next U	pdate So	rt Data Descer	nding
Messages V	3			
<	illi.			>
<td>id="d_emp orm&gt; ssages&gt;<td>" rowsPerPage essages&gt;</td><td>:="10"&gt;<td>dataVind 🔨</td></td></td>	id="d_emp orm> ssages> <td>" rowsPerPage essages&gt;</td> <td>:="10"&gt;<td>dataVind 🔨</td></td>	" rowsPerPage essages>	:="10"> <td>dataVind 🔨</td>	dataVind 🔨
<				
				2

- 3 If not selected, click the **messages** control in the design pane, and in the **Properties** view, select the **Quick Edit** tab.
- 4 In the **ID** field, enter retrieveMsg.

Quick Edit	ID:	retrieveMsg	8
Attributes	Layout:		•
	Global Only:		<b>•</b>
	Style:		Edit 👒

- 5 To add a DataWindow object retrieve listener, right-click the Employee ID DataWindow in the *employee.jsp* page, and select Add Listener RetrieveListener from the context menu.
- 6 In the **Create retrieveListener** dialog box, click **Type** to create a new Java class.

The New Java Class dialog box opens.

- 7 In the **Source Folder** field, make sure it reads *Tutorial**src*.
- 8 In the **Package** field, enter com.sybase.webapp.tutorial.
- 9 In the Name field, enter MyRetrieveListener.
- 10 Accept the remaining default settings and click **Finish** to create the Java class and return to the **Create retrieveListener** dialog box.

🔷 New Java Cla	ss	×
Java Class Create a new Java	ı dass.	P
Source fol <u>d</u> er:	Tutorial/src	Browse
Package:	com.sybase.webapp.tutorial	Bro <u>w</u> se
Enclosing type:		Browse
Na <u>m</u> e: Modifiers:	MyRetrieveListener]       If public       C default       C priyate       C abstract       F final       □ statig	
Superclass:	java.lang.Object	Browse
Interfaces:	Com.sybase.datastore.event.RetrieveListener	<u>A</u> dd
		Remove
Which method stub	y would you like to create? public static void main(String[] args) ✓	ect?
	Einish	Cancel

11 In the **Create retrieveListener** dialog box, click **OK** to create the MyRetrieveListener interface and add it to the Interfaces list.

Next, you are going to replace the default generated code in the Web Page editor with provided sample code that enables the display of messages in the Web browser as data is retrieved from the database.

12 Select File|Open File from the main menu bar to open the following file:

<installation directory>\sybase_workspace\web_development\eclipse \plugins\com.sybase.stf.jmt.template_1.5.0\tutorial\src\com\sybase \webapp\tutorial\MyRetrieveListener.java

- 13 Copy the contents of the Sybase-provided *MyRetrieveListener.java* file and replace them with the existing contents of the *MyRetrieveListener.java* file in the Web Page editor, and then close the Sybase-provided Java file.
- 14 Select **File**|**Save** from the menu bar to save the *MyRetrieveListener.java* file.
- Testing the Web page on the Tomcat 5.0 server
  - 1 If necessary, save the *employee.jsp* file.
  - 2 In the **WorkSpace Navigator**, right-click *employee.jsp* and select **Run** from the context menu.
  - 3 Select Choose an existing server, select Tomcat v.5.0Server @ local host from the list, and click Finish.

The Apache Tomcat server starts, and the JSP Page Template opens.

4 To test the Web page, click the **Previous** or **Next** button at the bottom of the window.

employee.jsp	D MyRetrieveLi	stener.java 🔗	JSF Page Template	× - 6
\$ = \$	http://localhost:202	20/Tutorial/employee	.faces	• • •
Employee ID	First Name	Last Name	Start Date	Birth Dat
102	Fran	Whitney	08/28/1984	06/05/1958
105	Matthew	Cobb	01/01/1985	12/04/1960
129	Philip	Chin	02/03/1985	10/30/1966
148	Julie	Jordan	04/05/1985	12/13/1951
160	Robert	Breault	06/17/1985	05/13/1947
184	Melissa	Espinoza	10/18/1985	12/14/1939
191	Jeannette	Bertrand	11/19/1985	12/21/1964
195	Marc	Dill	12/06/1985	07/19/1963
207	Jane	Francis	02/03/1986	09/12/1954
243	Natasha	Shishov	06/07/1986	04/22/1949
Previous	Next U	pdate Sor	rt Data Descen	ding
> >>  p;	age 1 of 8	<b>₽</b>		
				>

5 Before continuing, click the **Stop the server** icon in the **Servers** view to stop the server.

<u>.</u>
the cerve
yrichronize

Now you are ready to add client-side events to a DataWindow object, as described in "Lesson 2: Add client-side events," below.

### Lesson 2: Add client-side events

Client-side events and methods enable you to create Web pages with dynamic content without a round trip to the server, which can improve performance. During runtime, the runtime engine executes these events.

In this lesson, add a client-side event to your DataWindow object. This clientside event enables you to sort employee information in descending order by employee ID.

#### Adding client-side events to a DataWindow object

1 Display the *employee.jsp* page in the Web Page editor.

2 In the Web Page editor design pane, right-click the Employee ID DataWindow object, and select **Edit Client Side Event Scripts** from the context menu.



Next, edit the client-side event script to sort employee records in descending order when you click the **Sort Data Descending** button on the DataWindow object in the browser.

- 3 Select buttonClicked(row,objectName).
- 4 Enter the following JavaScript in the editor box and click **OK**.

```
var gobn = objectName;
if (gobn == "b_4") {
  d_emp.SetSort("emp_id D");
  d_emp.Sort();
}
if (gobn == "b 5") {
```
```
d_emp.SetSort("emp_id A");
d_emp.Sort();
}
Edit Client Side Event Scripts
```

ataWindow			
<pre>var gobn == objectName; if (gobn == 'b_1''){     d_emp_SetSort('mm_jid D'');     d_emp_SetSort('arm_jid A'');     d_emp_SetSort('arm_jid A'');     d_emp_SetSort('arm_jid A''); }</pre>	T		
	etaWindow yar gobn = objectName; f (gobn == "b_1")( d_emp.SotSotTemp.jd D"); d_emp.Sot(); f (gobn == "b_5"){ d_emp.Sot(); d_emp.Sot(); }	<pre>staWindow  var gobn = objectName; f(gobn == 'b_4')(     d_emp_SetSoft("emp_id D");     d_emp_SetSoft("emp_id A");     d_emp_SetSoft("emp_id A");     d_emp_Soft(); } T</pre>	<pre>staWindow  var gobn = objectName; if (gobn == "b_1"); d_smp_Sot(C'rem_jid D'); d_smp_Sot(C); } if (gobn == "b_5"); d_smp_Sot(Sot("emp_jid A"); d_smp_Sot(); } T</pre>

The code for the client-side events display in the Web Page editor.

a employee isb	×						
🛋 http://sybase	e.com/jsf 🖭 directiv	/e.page 🚉 http://jav	/a.sun.com/jsf/html	🛋 http://java.su	n.com/jsf/core		
HEAD							
Employee ID	First Name	Last Name	Start Date	Birth Date	Salary	Dept	
1000	emp_fname	emp_Iname	00/00/0000	00/00/0000	\$100.00	1000	<b>•</b>
Previous	Next U	pdate Sor	rt Data Descen	iding			
- Д							
nessages 🗊							
<h:fo< td=""><td>rm&gt; syb:dataWind</td><td>ow</td><td></td><td></td><td></td><td></td><td><u>••</u></td></h:fo<>	rm> syb:dataWind	ow					<u>••</u>
if (gobn d_emp.Se d_emp.So	<pre>dataWindo id="d_emp buttonClis == "b_4") ( :tSort("emp_is prt();</pre>	ring="dba/sql wObject="d_em " rowsPerPage cked='var gob d D");	/com.sybase.j p" libraryLis ="10" n = objectNar	jdbc2.jdbc. st="/WEB-IN me;	SybDriver F/pb/test	/jdbc:sybase .pbl"	≥:Tds:localhos
if (gobn d_emp.Se d_emp.So ) if (gobn d_emp.Se	<pre>dataWindo id="d emp buttonClin == "b_4")( etSort("emp_i ort(); == "b_5")( etSort("emp_in</pre>	ring="dba/sql wobject="d_em " rowsPerPage cked='var gob d D"); d &");	/com.sybase.j p" libraryLis ="10" n = objectNar	jdbc2.jdbc. st="/WEB-IN ne;	SybDriver F/pb/test	/jdbc:sybase .pbl" ]	≥:Tds:localhos
if (gobn d_emp.Se d_emp.So ) if (gobn d_emp.Se d_emp.So ) \>	<pre>dataWindo id="d_emp buttonClin == "b_4")( :tSort("emp_in ort(); == "b_5")( :tSort("emp_in rt();</pre>	ring="dba/sql wobject="d_emm" rowsPerPage cked='var gob d D"); d &");	/com.sybase. p" libraryLis ="10" n = objectNar	jdbe2.jdbe. st="/WEB-IN ne;	SybDriver F/pb/test	/jdbc:sybase .pbl" ]	e:Tds:localhos
if (gobn d_emp.Se d_emp.So ) if (gobn d_emp.Se d_emp.So d_emp.So d_emp.So 	<pre>dataWindo id="d_emp buttonClil == "b_4")( ttSort("emp_i"); ttSort("emp_i"); ttSort("emp_i"); ttSort("emp_i"); ttSort("emp_i");</pre>	<pre>ring="dbaysgl vobject="d_em " rowsPerPage cked='var gob d D"); d A"); type="com.sy d="retrieveMs</pre>	/com.sybase. p" libraryLin =r10" n = objectNar base.webapp.t g"> <td>jdbe2.jdbe. st="/WEB-IN we; putorial.My yes&gt;</td> <td>SybDriver F/pb/test RetrieveL</td> <td>/jdbc:sybase .pbl" ] .istener"&gt;<td>e:Tds:localhos</td></td>	jdbe2.jdbe. st="/WEB-IN we; putorial.My yes>	SybDriver F/pb/test RetrieveL	/jdbc:sybase .pbl" ] .istener"> <td>e:Tds:localhos</td>	e:Tds:localhos
if (gobn d_emp.Se d_b) if (gobn d_emp.Se d_emp.Se d_emp.So )'> <syb:retr <td><pre>dataWindo id="d_emp buttonClil == "b_4")( ttSort("emp_i rt(); == "b_5")( ttSort("emp_i rt(); rieveListener thrmessages in form&gt;</pre></td><td><pre>ring="dbaysgl vobject="d_em " rowsPerPage cked='var gob d D"); d A"); type="com.sy d="retrieveNs</pre></td><td>/com.sybase. p" libraryLin =r10" n = objectNar base.webapp.t g"&gt;<td>jdbe2.jdbe. st="/WEB-IN we; cutorial.My ges&gt;</td><td>SybDriver F/pb/test RetrieveL</td><td>/jdbc:sybase .pbl" </td><td>e:Tds:localhos</td></td></syb:retr 	<pre>dataWindo id="d_emp buttonClil == "b_4")( ttSort("emp_i rt(); == "b_5")( ttSort("emp_i rt(); rieveListener thrmessages in form&gt;</pre>	<pre>ring="dbaysgl vobject="d_em " rowsPerPage cked='var gob d D"); d A"); type="com.sy d="retrieveNs</pre>	/com.sybase. p" libraryLin =r10" n = objectNar base.webapp.t g"> <td>jdbe2.jdbe. st="/WEB-IN we; cutorial.My ges&gt;</td> <td>SybDriver F/pb/test RetrieveL</td> <td>/jdbc:sybase .pbl" </td> <td>e:Tds:localhos</td>	jdbe2.jdbe. st="/WEB-IN we; cutorial.My ges>	SybDriver F/pb/test RetrieveL	/jdbc:sybase .pbl" 	e:Tds:localhos
if (gobn d_emp.Se d_emp.So ) if (gobn d_emp.Se d_emp.So ) ) syb:retr 	<pre>dataWindo id="d emp puttonClii == "b_4")( ttSort("emp_i rt(); == "b_5")( ttSort("emp_i rt(); rieveListener threessages in oom&gt;</pre>	<pre>ring="dbaysgl vibject="d_em " rowsPerPage cked='var gob d D"); d A"); type="com.sy d="retrieveHs</pre>	/com.sybase. p" libraryLis =~10" n = objectNar base.webapp.t g"> <td>jdbc2.jdbc. st="/WEB-IN we; cutorial.My ges&gt;</td> <td>SybDriver F/pb/test RetrieveL</td> <td>/jdbc:sybase .pbl" </td> <td>e:Tds:localhos</td>	jdbc2.jdbc. st="/WEB-IN we; cutorial.My ges>	SybDriver F/pb/test RetrieveL	/jdbc:sybase .pbl" 	e:Tds:localhos
if (gobn d_emp.Se d_emp.Se d_emp.Se d_emp.Se d_emp.Se d_emp.Se  	<pre>dat&amp;Windo id="d emp puttonCli == "b_4")( ttSort("emp_i rt(); == "b_5")( ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort("emp_i rt(); ttSort(); ttSort("emp_i rt(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort(); ttSort();</pre>	<pre>ring="dba/sql vobject="d_em " rowsPerPage cked='var gob d D"); d A"); type="com.sy d="retrieveMs</pre>	/com.sybase.p p" libraryLis ="10" n = objectNar base.webapp.t g"> <td>dbe2.jdbe. st="/WEB-IN we; vutorial.My ges&gt;</td> <td>SybDriver F/pb/test RetrieveL</td> <td>/jdbc:sybase .pbl"</td> <td>e:Tds:localhos</td>	dbe2.jdbe. st="/WEB-IN we; vutorial.My ges>	SybDriver F/pb/test RetrieveL	/jdbc:sybase .pbl"	e:Tds:localhos

### * Testing the employee.jsp Web page on the Tomcat 5.0 server

- 1 Select **File**|**Save** from the menu bar to save the *employee.jsp* file.
- 2 Right-click the design pane and select **Run** from the context menu.
- 3 Select **Choose existing server**, select **Tomcat v5.0Server** @ **localhost** from the list, and click **Finish**.

The Apache Tomcat server starts, and the JSF Page Template opens.

4 Click the **Sort Data Descending** button on the Web page to sort the clientside event data in a descending order by Employee ID.

Employee ID	First Name	Last Name	Start Date	Birth Date	Salary	Dept	
1751	Alex	Ahmed	07/12/1994	12/12/1963	\$34,992.00	400	•
1740	Robert	Nielsen	06/24/1994	06/19/1965	\$34,889.00	400	1
1684	Janet	Hildebrand	03/15/1994	10/31/1955	\$45,829.00	400	`
1658	Michael	Lynch	02/27/1994	01/18/1973	\$24,903.00	500	2
1643	Elizabeth	Lambert	12/15/1993	09/12/1968	\$29,384.00	400	•
1615	Sheila	Romero	11/19/1993	09/12/1972	\$27,500.00	500	1
1607	Mark	Morris	10/13/1993	01/08/1941	\$61,300.00	400	`
1596	Catherine	Pickett	08/12/1993	11/18/1959	\$47,653.00	200	\$
1576	Scott	Evans	07/01/1993	11/15/1960	\$68,940.00	400	1
1570	Anthony	Rebeiro	05/29/1993	04/12/1963	\$34,576.00	500	1
) >>) p ginRetrieve.	age 1 of 8 Event.getRetActic	pn() = com sybas	e. datastore. enur	ns.BeginRetrie	veAction@	ee6ad6#	

5 Before continuing, click the **Stop the server** icon in the **Servers** view to stop the server.

## CHAPTER 7 Debugging Web Applications

The chapter introduces the Debug perspective in Sybase WorkSpace. In this tutorial, you will set breakpoints in your Web application code, which causes Sybase WorkSpace to launch the Debug perspective.

When you create your own Web applications, the Debug perspective and its associated views enables you to step through the breakpoints, identify compilation errors in the Java source code, view error logs, and perform other basic debugging tasks.

## **Debug perspective**

Throughout the development cycle, you will use Web Application Development tooling to debug your Web pages. You can debug code by setting breakpoints on the Web page or in Java managed beans by setting breakpoints in the Java class.

Before you can perform the debugging tutorials, you must complete all the previous tutorials in the guide.

#### Setting breakpoints and launching the Debug perspective

**Note** Before you begin the debugging process, be sure that the Tomcat server is stopped.

- 1 In the **WorkSpace Navigator**, double-click the *login.jsp* file to open the file in the Web Page editor.
- 2 To set a breakpoint in the Web Page editor, double-click in the vertical gray border to the left of the line following the *<h:panelGrid>* tag:

```
<h:outputText value="Login ID"></h:outputText>
```

A blue dot in the gray border indicates that you successfully set the breakpoint.

🔄 login. jsj	X		- 8
e directi	ve.page 🗮 http://java.sun.com/jsf/h	tml 🗮 http://java.sun.com/jsf/core	
Login I	D #{loginID}		=
Passwo	yrd *******		-
Login			~
< <body <f:vi< td=""><td>(META HTTP-EQUIV="Conten (TITLE&gt;JSF Page Template UD&gt; /&gt; Lew&gt;</td><td>t-Type" CONTENT="text/html; </td><td>CHARSET=U</td></f:vi<></body 	(META HTTP-EQUIV="Conten (TITLE>JSF Page Template UD> /> Lew>	t-Type" CONTENT="text/html; 	CHARSET=U
<h:fo< td=""><td>)rm&gt;</td><td>&lt;</td><td></td></h:fo<>	)rm>	<	
	<pre><h:outputtext value="&lt;br"><h:outputtext id"="" login="" value="&lt;br&gt;&lt;h:outputText value=&lt;br&gt;&lt;h:inputSecret value&lt;br&gt;&lt;/h:panelGrid&gt;&lt;/pre&gt;&lt;/th&gt;&lt;th&gt;&lt;pre&gt;"></h:outputtext> #{loginBean.loginID}"&gt;</h:outputtext> ="#{loginBean.password)"&gt; </pre>	nputText> h:inputSe	
<			>

Next, set a breakpoint in the loginBean.

- 3 In the **Data Bindings** view, right-click **loginBean** and select **Open** from the context menu.
- 4 In the source view of the Web Page editor, set a breakpoint by doubleclicking in the gray border to the left of the line following the *if* statement inside the validateLogin() method.

```
if (this.loginID.equals("sybase") &&
(this.password.equals("sybase")))
```

A blue dot appears in the gray border identifying the breakpoint.



5 Return to *login.jsp* page, right-click the design pane, and select **Debug** from the context menu to debug the Web page.

н		incipity) a ratio			1
Log	in ID #{loginID}				III
Pass	word *******	🖓 Undo	Ctrl+Z		-
1.00	in a	🍄 Redo	Ctrl+Y		~
209	(TITHE/OST TAGE TEMPTAGE ()	of Cut	Ctrl+X		~
?</td <td>HEAD&gt;</td> <td>Copy</td> <td>Ctrl+C</td> <td></td> <td></td>	HEAD>	Copy	Ctrl+C		
<b< td=""><td>ODY&gt;</td><td>💼 Paste</td><td>Ctrl+V</td><td></td><td></td></b<>	ODY>	💼 Paste	Ctrl+V		
<f< td=""><td>:view&gt;</td><td>💢 Delete</td><td>Delete</td><td></td><td></td></f<>	:view>	💢 Delete	Delete		
<h< td=""><td>:form&gt;</td><td></td><td></td><td>-</td><td>ai.</td></h<>	:form>			-	ai.
	<h:panelgrid columns="2"></h:panelgrid>	Alian	ormac		
1	<h:inputtext p<="" td="" value="#{&lt;/td&gt;&lt;td&gt;Text Style&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;h:input]&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;h:outputText value="><td>Make Link</td><td></td><td>xt&gt;</td><td></td></h:inputtext>	Make Link		xt>	
	<h:inputsecret value="&lt;/td&gt;&lt;td&gt;0.&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;- "><td></td></h:inputsecret>				
			_		
	<h:commandButton value="Lo</td> <td>🐞 Debug</td> <td></td> <td>ean.valid</td> <td></td>	🐞 Debug		ean.valid	
		Show View [®]			~

6 In the **Debug on Server** wizard, select **Choose an existing server**, select **Tomcat v5.0Server** @ **localhost**, and click **Finish**.

The Confirm Perspective Switch dialog box displays, which confirms that you want to change to the Debug perspective.

♦ Confirm Perspective Switch	×
This kind of launch is configured to open the Debug perspective when suspends. Do you want to open this perspective now?	it
Remember my decision	
Yes	

7 Click **Yes** to open the Debug perspective.



In the Debug perspective, you can run through the breakpoints and perform debugging tasks for your Web applications. You have now completed the Web Application Development component tutorial.

# Available debugging tools

	These tools can assist you in troubleshooting design and development problems:
Problems view	Refer to the Problems view to identify compilation errors in the Java source code of a JSP page.
Errors markers	Use the errors markers that appear on the vertical ruler of the source view in the Web Page editor to identify the cause and solution for an error. Move your mouse over the error marker to display the problem cause; double-click the error marker to display possible resolutions.
Error Log view	Review error logs to identify design and runtime errors.
	Select <b>Windows</b>  Show View Others from the menu bar, and then select PDE Runtime Error Log from the list.