



Artix™

Using Artix Designer

Version 4.1, September 2006

IONA Technologies PLC and/or its subsidiaries may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this publication. Except as expressly provided in any written license agreement from IONA Technologies PLC, the furnishing of this publication does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Any rights not expressly granted herein are reserved.

IONA, IONA Technologies, the IONA logos, Orbix, Artix, Making Software Work Together, Adaptive Runtime Technology, Orbacus, IONA University, and IONA XMLBus are trademarks or registered trademarks of IONA Technologies PLC and/or its subsidiaries.

Java and J2EE are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. CORBA is a trademark or registered trademark of the Object Management Group, Inc. in the United States and other countries. All other trademarks that appear herein are the property of their respective owners.

While the information in this publication is believed to be accurate, IONA Technologies PLC makes no warranty of any kind to this material including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. IONA shall not be liable for errors contained herein, or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

COPYRIGHT NOTICE

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photocopying, recording or otherwise, without prior written consent of IONA Technologies PLC. No third-party intellectual property right liability is assumed with respect to the use of the information contained herein. IONA Technologies PLC assumes no responsibility for errors or omissions contained in this publication. This publication and features described herein are subject to change without notice.

Copyright © 1999-2006 IONA Technologies PLC. All rights reserved.

All products or services mentioned in this publication are covered by the trademarks, service marks, or product names as designated by the companies that market those products.

Updated: October 3, 2006

Contents

List of Figures	5
Preface	7
What is Covered in this Book	7
Who Should Read this Book	7
How to Use this Book	7
The Artix Library	8
Getting the Latest Version	11
Searching the Artix Library	11
Artix Online Help	11
Artix Glossary	12
Additional Resources	12
Document Conventions	12
Chapter 1 Getting Started with Artix Designer	15
Introducing Artix Designer	16
Installing Artix Designer	17
Launching Artix Designer	18
Setting Artix Designer Preferences	20
The Artix Designer Perspectives	21
Using the Online Help and Cheat Sheets	23
Importing and Running the Artix Demos	25
Chapter 2 Creating Projects	27
Creating Projects	28
Creating Basic Projects	29
Creating Other Project Types	31
Importing Projects	32
Chapter 3 Generating WSDL	33
Creating an Empty WSDL File	34
Importing and linking to WSDL files	35
Generating WSDL from Different Sources	38

Working with XML Schema Files	39
Other WSDL-related Tasks	40
Chapter 4 Using the WSDL Editor	43
Introducing the WSDL Editor	44
Using the WSDL Editor Wizards	46
Setting WSDL Editor Preferences	47
Chapter 5 Generating, Launching, and Deploying Artix Solutions	49
Introducing Launch Configurations	50
Generating Code	53
Launching Artix Applications	55
Deploying a Service into a Container	57
Launching a Container	58
Testing a Service	60
Chapter 6 Exposing a Database as a Web Service	63
Creating a Database Web Services Project	64
Adding Operations to the DB Config File	65
Generating and running the Web service	68
Chapter 7 Managing Services and Containers	71
Introducing the Container Admin perspective	72
Container Admin Tasks	75
Chapter 8 Securing Artix Applications	77
Introducing Artix Security	78
Creating secure projects	79
Index	81

List of Figures

Figure 1: The Artix Designer Welcome Page	19
Figure 2: Artix Designer Preferences	20
Figure 3: The Open Perspective icon	21
Figure 4: The Cheat Sheet Selection Window	24
Figure 5: Linking to a WSDL File	36
Figure 6: The Linked WSDL File Icon	37
Figure 7: The Diagram View in the WSDL Editor	44
Figure 8: The Outline View	45
Figure 9: Artix Designer Preferences—WSDL Settings	47
Figure 10: Artix Designer Preferences—Validation Settings	48
Figure 11: The Artix Tools Window	50
Figure 12: Launching Java Applications from the Run Window	51
Figure 13: Launching Applications from the External Tools Window	52
Figure 14: Running a Java Application	55
Figure 15: Running a C++ Application	56
Figure 16: The Test Results View with an Actual Result Displayed	60
Figure 17: Actual and Expected Results where the Test was Successful	61
Figure 18: Actual and Expected Results where the Test Failed	61
Figure 19: The DB Config Editor	65
Figure 20: Adding Parameters to a Stored Procedure Operation	66
Figure 21: The Generate WSDL/Service Wizard	68
Figure 22: The Service Explorer View	72
Figure 23: The Services List View	73
Figure 24: The Admin Events View	73
Figure 25: The Properties view	74
Figure 26: The Security Tabbed Page in the Artix Tools Window	80

LIST OF FIGURES

Preface

What is Covered in this Book

This book aims to provide an overview of the functionality available in IONA Artix Designer, the Eclipse-based tool for designing Artix solutions.

It does not set out to provide a detailed description of all the tasks that you can perform in each application. For step-by-step instructions, see the Artix Designer online help.

Who Should Read this Book

This book is intended for anyone who plans to use Artix Designer to design, develop, and test Artix solutions.

How to Use this Book

This book is structured as follows:

- [Chapter 1](#) provides a brief introduction to Artix Designer
- [Chapter 2](#) gives an overview of the different kinds of Artix Designer project that you can create.
- [Chapter 3](#) explains how to generate WSDL from a variety of sources
- [Chapter 4](#) gives an overview of how to use the WSDL Editor to add WSDL element to your Artix contract.
- [Chapter 5](#) shows how you to generate, deploy, launch, and test Artix Web services.
- [Chapter 6](#) gives an outline of how to expose a database as a Web service using Artix Designer
- [Chapter 7](#) describes the functionality available in the Container Admin perspective

The Artix Library

The Artix documentation library is organized in the following sections:

- [Getting Started](#)
- [Designing Artix Solutions](#)
- [Configuring and Managing Artix Solutions](#)
- [Using Artix Services](#)
- [Integrating Artix Solutions](#)
- [Integrating with Management Systems](#)
- [Reference](#)
- [Artix Orchestration](#)

Getting Started

The books in this section provide you with a background for working with Artix. They describe many of the concepts and technologies used by Artix. They include:

- [Release Notes](#) contains release-specific information about Artix.
- [Installation Guide](#) describes the prerequisites for installing Artix and the procedures for installing Artix on supported systems.
- [Getting Started with Artix](#) describes basic Artix and WSDL concepts.
- [Using Artix Designer](#) describes how to use Artix Designer to build Artix solutions.
- [Artix Technical Use Cases](#) provides a number of step-by-step examples of building common Artix solutions.

Designing Artix Solutions

The books in this section go into greater depth about using Artix to solve real-world problems. They describe how to build service-oriented architectures with Artix and how Artix uses WSDL to define services:

- [Building Service-Oriented Infrastructures with Artix](#) provides an overview of service-oriented architectures and describes how they can be implemented using Artix.
- [Writing Artix Contracts](#) describes the components of an Artix contract. Special attention is paid to the WSDL extensions used to define Artix-specific payload formats and transports.

Developing Artix Solutions

The books in this section how to use the Artix APIs to build new services:

- [Developing Artix Applications in C++](#) discusses the technical aspects of programming applications using the C++ API.
- [Developing Advanced Artix Plug-ins in C++](#) discusses the technical aspects of implementing advanced plug-ins (for example, interceptors) using the C++ API.
- [Developing Artix Applications in Java](#) discusses the technical aspects of programming applications using the Java API.

Configuring and Managing Artix Solutions

This section includes:

- [Configuring and Deploying Artix Solutions](#) explains how to set up your Artix environment and how to configure and deploy Artix services.
- [Managing Artix Solutions with JMX](#) explains how to monitor and manage an Artix runtime using Java Management Extensions.

Using Artix Services

The books in this section describe how to use the services provided with Artix:

- [Artix Router Guide](#) explains how to integrate services using the Artix router.
- [Artix Locator Guide](#) explains how clients can find services using the Artix locator.
- [Artix Session Manager Guide](#) explains how to manage client sessions using the Artix session manager.
- [Artix Transactions Guide, C++](#) explains how to enable Artix C++ applications to participate in transacted operations.
- [Artix Transactions Guide, Java](#) explains how to enable Artix Java applications to participate in transacted operations.
- [Artix Security Guide](#) explains how to use the security features in Artix.

Integrating Artix Solutions

The books in this section describe how to integrate Artix solutions with other middleware technologies.

- [Artix for CORBA](#) provides information on using Artix in a CORBA environment.
- [Artix for J2EE](#) provides information on using Artix to integrate with J2EE applications.

For details on integrating with Microsoft's .NET technology, see the documentation for Artix Connect.

Integrating with Management Systems

The books in this section describe how to integrate Artix solutions with a range of enterprise and SOA management systems. They include:

- [IBM Tivoli Integration Guide](#) explains how to integrate Artix with the IBM Tivoli enterprise management system.
- [BMC Patrol Integration Guide](#) explains how to integrate Artix with the BMC Patrol enterprise management system.
- [CA-WSDM Integration Guide](#) explains how to integrate Artix with the CA-WSDM SOA management system.
- [AmberPoint Integration Guide](#) explains how to integrate Artix with the AmberPoint SOA management system.

Reference

These books provide detailed reference information about specific Artix APIs, WSDL extensions, configuration variables, command-line tools, and terms. The reference documentation includes:

- [Artix Command Line Reference](#)
- [Artix Configuration Reference](#)
- [Artix WSDL Extension Reference](#)
- [Artix Java API Reference](#)
- [Artix C++ API Reference](#)
- [Artix .NET API Reference](#)
- [Artix Glossary](#)

Artix Orchestration

These books describe the Artix support for Business Process Execution Language (BPEL), which is available as an add-on to Artix. These books include:

- [Artix Orchestration Release Notes](#)
- [Artix Orchestration Installation Guide](#)
- [Understanding Artix Orchestration](#)
- [Artix Orchestration Administration Console Help](#).

Getting the Latest Version

The latest updates to the Artix documentation can be found at <http://www.iona.com/support/docs>.

Compare the version dates on the web page for your product version with the date printed on the copyright page of the PDF edition of the book you are reading.

Searching the Artix Library

You can search the online documentation by using the **Search** box at the top right of the documentation home page:

<http://www.iona.com/support/docs>

To search a particular library version, browse to the required index page, and use the **Search** box at the top right, for example:

<http://www.iona.com/support/docs/artix/4.1/index.xml>

You can also search within a particular book. To search within a HTML version of a book, use the **Search** box at the top left of the page. To search within a PDF version of a book, in Adobe Acrobat, select **Edit|Find**, and enter your search text.

Artix Online Help

Artix Designer and Artix Orchestration Designer include comprehensive online help, providing:

- Step-by-step instructions on how to perform important tasks
- A full search feature
- Context-sensitive help for each screen

There are two ways that you can access the online help:

- Select **Help|Help Contents** from the menu bar. The help appears in the contents panel of the Eclipse help browser.
- Press **F1** for context-sensitive help.

In addition, there are a number of cheat sheets that guide you through the most important functionality in Artix Designer and Artix Orchestration Designer. To access these, select **Help|Cheat Sheets**.

Artix Glossary

The [Artix Glossary](#) is a comprehensive reference of Artix terms. It provides quick definitions of the main Artix components and concepts. All terms are defined in the context of the development and deployment of Web services using Artix.

Additional Resources

The [IONA Knowledge Base](http://www.iona.com/support/knowledge_base/index.xml) (http://www.iona.com/support/knowledge_base/index.xml) contains helpful articles written by IONA experts about Artix and other products.

The [IONA Update Center](http://www.iona.com/support/updates/index.xml) (<http://www.iona.com/support/updates/index.xml>) contains the latest releases and patches for IONA products.

If you need help with this or any other IONA product, go to [IONA Online Support](http://www.iona.com/support/index.xml) (<http://www.iona.com/support/index.xml>).

Comments, corrections, and suggestions on IONA documentation can be sent to docs-support@iona.com.

Document Conventions

Typographical conventions

This book uses the following typographical conventions:

Fixed width

Fixed width (Courier font) in normal text represents portions of code and literal names of items such as classes, functions, variables, and data structures. For example, text might refer to the `IT_Bus::AnyType` class.

Constant width paragraphs represent code examples or information a system displays on the screen. For example:

```
#include <stdio.h>
```

Fixed width italic

Fixed width italic words or characters in code and commands represent variable values you must supply, such as arguments to commands or path names for your particular system. For example:

```
% cd /users/YourUserName
```

Italic

Italic words in normal text represent *emphasis* and introduce *new terms*.

Bold Bold words in normal text represent graphical user interface components such as menu commands and dialog boxes. For example: the **User Preferences** dialog.

Keying Conventions

This book uses the following keying conventions:

No prompt	When a command's format is the same for multiple platforms, the command prompt is not shown.
%	A percent sign represents the UNIX command shell prompt for a command that does not require root privileges.
#	A number sign represents the UNIX command shell prompt for a command that requires root privileges.
>	The notation > represents the MS-DOS or Windows command prompt.
...	Horizontal or vertical ellipses in format and syntax descriptions indicate that material has been eliminated to simplify a discussion.
[]	Brackets enclose optional items in format and syntax descriptions.
{ }	Braces enclose a list from which you must choose an item in format and syntax descriptions.
	In format and syntax descriptions, a vertical bar separates items in a list of choices enclosed in { } (braces). In graphical user interface descriptions, a vertical bar separates menu commands (for example, select File Open).

Getting Started with Artix Designer

This chapter discusses the basics of using Artix Designer.

In this chapter

This chapter discusses the following topics:

Introducing Artix Designer	page 16
Installing Artix Designer	page 17
Launching Artix Designer	page 18
Setting Artix Designer Preferences	page 20
The Artix Designer Perspectives	page 21
Using the Online Help and Cheat Sheets	page 23
Importing and Running the Artix Demos	page 25

Introducing Artix Designer

Overview

Artix Designer is a development environment that provides a unified workspace for building Artix Web services solutions.

It ships as a collection of plug-ins to the Eclipse open source development platform. When you install Artix Designer as part of an Artix kit, you install Eclipse, including its integrated development environments (IDEs) for both C++ and Java, at the same time.

Getting to know Eclipse

If you are not familiar with Eclipse, you can find out more at www.eclipse.org.

We also recommend that you take time to familiarize yourself with the basics of the Eclipse platform by studying the Workbench User Guide and, if you are a Java developer, the Java Development Guide, in the Eclipse online help.

You may also want to take one or more of the Eclipse cheat sheets, which are accessible from the **Help** menu. Cheat sheets are a standard Eclipse feature that guide you through some of the platform's application development processes.

Artix Designer features

You can use Artix Designer to do the following:

- Create and edit Artix contracts based on Web Services Description Language (WSDL)
- Generate WSDL from a range of sources, including CORBA IDL, EJB session beans, Java classes, and COBOL copybook files
- Generate applications and service plug-ins in C++ and Java
- Design, create, and deploy enterprise integration solutions for z/OS
- Expose a database as a Web service
- Deploy service plug-ins into an Artix container
- Manage service plug-ins running in an Artix container
- Test services

Installing Artix Designer

Supported platforms

The Artix Designer plug-ins ship as part of the Windows, Linux, and Solaris versions of the Artix kit only.

Installing Artix Designer

When you run the Artix installer, the Eclipse platform, Artix Designer is installed by default.

The installer adds an `eclipse` folder under the `InstallDir\artix\VersionNumber` directory, containing all the plug-ins needed to run the Eclipse platform and its associated IDEs, as well as the Artix Designer plug-ins.

Adding to an existing Eclipse installation

If you already have Eclipse installed on your machine, you may want to add the Artix Designer plug-ins to your existing installation. See the [Artix Installation Guide](#) for details.

Launching Artix Designer

Starting as part of Artix

If you installed the Artix Designer plug-ins using the Artix installer, you can launch it using the `start_eclipse` script in the `InstallDir/artix/VersionNumber/bin` directory.

Windows users can also start Artix Designer from the **IONA | Artix 4.1** sub-menu under **Start | Programs**.

Choosing a workspace

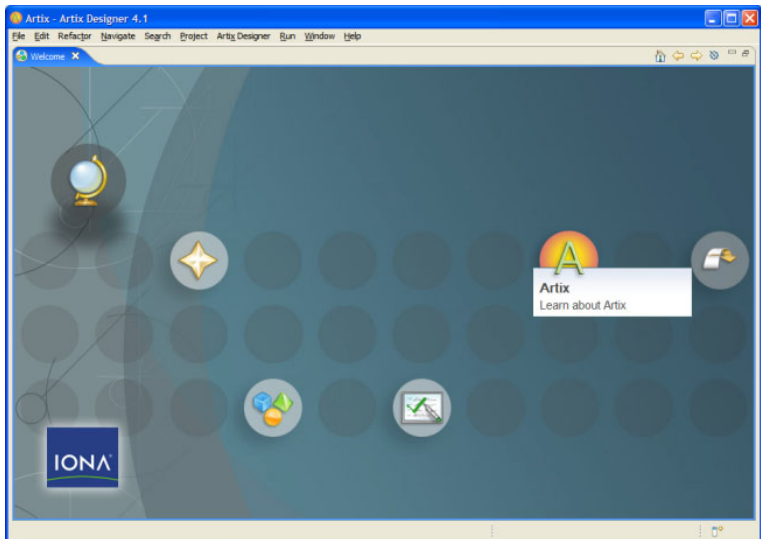
When you first launch Artix Designer, you are prompted to choose a directory for your Eclipse workspace. This is where all your Artix Designer projects will be stored.

It is best to store your workspace outside the Artix installation directory, to prevent it being deleted should you ever need to remove Artix from your machine—for example, when upgrading to a new version.

Exploring the Welcome screen

The Artix Designer Welcome page to provide you with some useful links and tips on how to get started with Eclipse and Artix Designer.

Figure 1: *The Artix Designer Welcome Page*



Click the **Learn about Artix** icon to go to a page that provides links to:

- The latest Artix release notes
- Artix Designer tutorials (cheat sheets)
- The Artix documentation library
- The IONA online support registration page

Setting Artix Designer Preferences

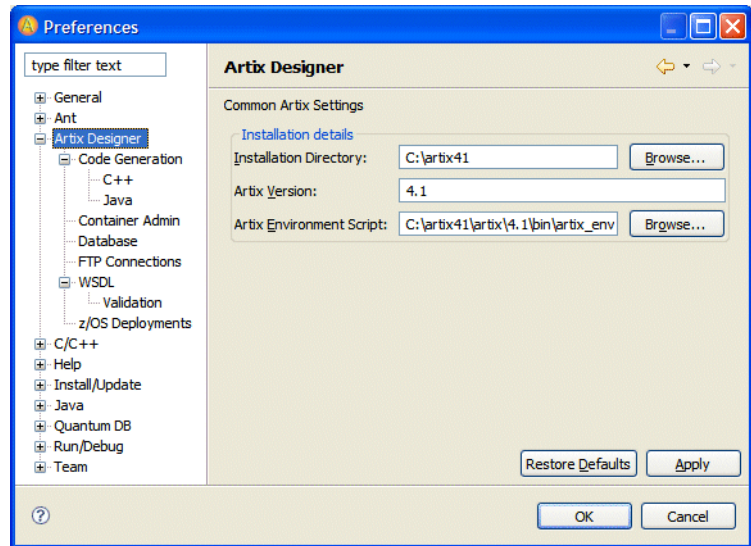
Opening the Artix Designer preferences panel

There are a number of preferences that you can set for Artix Designer.

To view the Artix Designer preferences panel:

1. From the **Window** menu, select **Preferences**.
2. Expand the **Artix Designer** node.

Figure 2: *Artix Designer Preferences*



For more details ...

For detailed help on each of the Artix Designer Preferences pages, select any of the Artix-related options in the Preferences window, and press **F1**.

The Artix Designer Perspectives

What is a perspective?

In Eclipse, perspectives define the initial view set and layout of the Workbench window. They provide the functionality needed for you to accomplish a specific task or to work with specific resources.

A perspective is made up of one or more views.

In Artix Designer, certain perspectives are associated with particular project types or Artix Tools configurations. You are prompted to switch perspective when you create an Artix Designer project or a container deployment configuration for the first time. You also have the option of always associating the relevant perspective with that project type or configuration.

Artix perspectives

Artix Designer ships with the following perspectives:

- Artix—Associated with basic, CORBA, and EJB Web services projects
 - Artix for z/OS—Associated with z/OS projects
 - Artix Database—Associated with Artix database projects
 - Artix Container Admin—Associated with the container deployment launch configuration
-

Switching perspectives

To switch perspective in Artix Designer, select **Window | Open Perspective** from the menu bar. If no Artix options appear in the sub-menu, select **Other** and then choose a perspective from the Select Perspective dialog box.

Alternatively, click the **Open Perspective** icon in the top right of the workbench.

Figure 3: *The Open Perspective icon*















Using the Artix Designer menu and toolbar

The options in the Artix Designer menu and toolbar allow you to perform common tasks in the application.

The following options are available:

Table 1: *Artix Designer Toolbar and Menu Options*

Button/Menu Option	Description
	Run the latest Artix Tools launch configuration ^a
	Import Artix demos
	Add <code>import</code> element to currently selected WSDL or XSD file
	Add <code>type</code> element to currently selected WSDL or XSD file
	Add <code>message</code> element to currently selected WSDL file
	Add <code>portType</code> element to currently selected WSDL file
	Add <code>binding</code> element to currently selected WSDL file
	Add <code>service</code> element to currently selected WSDL file
	Add <code>route</code> element to currently selected WSDL file
	Create an access control list for currently selected WSDL file
	Add SOAP service and binding to currently selected WSDL
	Add CORBA service and binding to currently selected WSDL

a. If a launch configuration already exists, clicking this button launches the last-used configuration. Click the down arrow next to the button to run other configurations or to open the Artix Tools window.

Using the Online Help and Cheat Sheets

Launching the online Help

Online help for Artix Designer is integrated into the Eclipse help system. To launch help in Eclipse, select **Help|Help Contents** from the menu bar. The Artix Designer help appears in the table of contents in the Eclipse help browser.

Viewing context-sensitive Help

To view context-sensitive help on a particular Artix Designer wizard or dialog box, press **F1**. Context-sensitive help opens either in a dynamic help view or in a pop-up window. In both cases, links to topics relating to the currently open window are provided.

Note: You can configure how context-sensitive help appears in Eclipse by going to the Help panel of the Preferences window.

Launching cheat sheets

Cheat sheets are a standard Eclipse feature that guide you through some of the platform's application development processes.

Each cheat sheet is designed to help you complete some task, and it lists the steps required to help you achieve that goal. As you progress from one step to the next, the cheat sheet automatically launches the required tools for you.

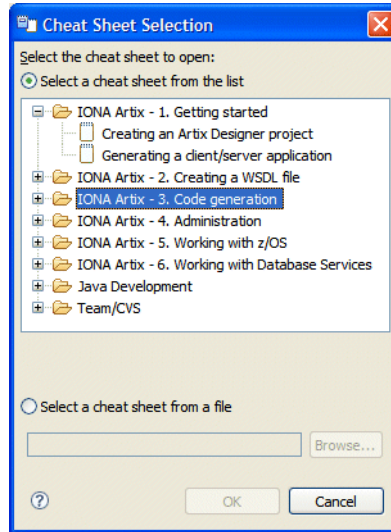
If there is a manual step in the process, the step tells you to perform the task and click a button in the cheat sheet to move on to the next step.

Artix Designer includes a number of cheat sheets covering the most important functionality in the product.

To access the Artix Designer cheat sheets:

1. Select **Help|Cheat Sheets**.
2. In the Cheat Sheet Selection window, expand one of the IONA Artix Designer folders and select a cheat sheet.

Figure 4: *The Cheat Sheet Selection Window*



3. Click **OK**.

Importing and Running the Artix Demos

Overview

Artix provides a number of examples that demonstrate how the product's features work and that can help you get started with Artix. The demos are located in:

```
InstallDir\artix\VersionNumber\demos
```

You can import a subset of these demos—the basic demos—into Artix Designer and run them from your workspace.

Before you begin

Before you import a demo, ensure that Artix Designer is correctly configured for compiling code automatically. See [“Compiling code automatically” on page 53](#) for details.

Note: The code is compiled using the Ant build system that accompanies the demos, and not the Eclipse JDT. This means you must have a valid JDK installed on your machine and your PATH environment variable set to include the `JDKInstallDir/bin` directory, so that Ant can find your Java compiler.

Importing a demo

To import a demo into Artix Designer, select **Artix Designer | Import Demos** and choose the demo that you want to import. You can choose to import the demo as either a Java or C++ project.

The demo is imported and the code is compiled automatically. Progress on the compilation process is displayed in the Console view.

Note: Although the demo directory and files appear in your Artix Designer workspace, they remain physically located in the `InstallDir\artix\VersionNumber\demos` directory.

Running a demo

To run a demo application or service:

1. Select **Run | External Tools | External Tools**.
2. In the External tools window, expand the **Program** node in the **Configurations** tree on the left and select the start script for the application or service that you want to run.
3. If you are running a client, add any necessary arguments to the **Arguments** field.
4. Click **Run**.

Server applications and Artix containers launch in a separate command line window. Client applications launch in the Console view.

Note: Launching a container from the External Tools window does not automatically add the container and service to the Container Admin perspective. If you want to manage a demo container in the Container Admin perspective, check the port number that the container is running on by opening the corresponding start script for the service, then add the container to the console.

Creating Projects

This chapter deals with the different project types available in Artix Designer.

In this chapter

This chapter discusses the following topics:

Creating Projects	page 28
Creating Basic Projects	page 29
Creating Other Project Types	page 31
Importing Projects	page 32

Creating Projects

What is a project?

In Eclipse, all development work is done within a project, so you must create a project before you can begin creating a WSDL file, XSD schema or Artix application.

Project directory structure

When you create a project in Artix Designer or any other Eclipse plug-in, you create a folder bearing the project name and containing a `.project` file.

Project types

In Artix Designer, you can create the following project types:

- Basic projects
- Projects from CORBA input
- Projects from EJB input
- Projects from database input
- z/OS projects

Creating Basic Projects

Project templates

When creating a basic Web services project in Artix Designer, you have the option of using a project template that allows you to automatically generate starting point code in either C++ or Java based on a single WSDL file.

The following project templates are available:

- Empty project—generates an empty directory structure and the necessary Eclipse `.project` file. Select this option if you want to write a WSDL file from scratch.
- Artix router—generates a start script and configuration file for deploying an Artix router inside an Artix container.
- C++/Java client/server/client and server—generates the starting point C++ or Java code, start scripts, and configuration files needed to implement a Web service application.
- C++/Java client and container—generates the starting point C++ or Java code for a service plug-in, an Artix container in which to deploy the service, and a client to run against the service.
- C++/Java container—generates the starting point C++ or Java code for a service plug-in and a container in which to deploy the service.

Creating a basic project

To create a basic Web services project in Artix Designer:

1. From the File menu select **New | Project**.
2. In the New Project dialog box, expand the **IONA Artix Designer** folder.
3. Click **Basic Web Services Project** and then click **Next**.
4. In the General Details panel, enter the name of the Web services project in the **Project name** field.
5. Select a template from the **Project templates** list.
6. You now have a number of choices:
 - ◆ If you selected **Empty project**, click **Finish**.
 - ◆ If you selected **Artix Router**, click **Browse** to select the WSDL file from which the router configuration will be generated.

- ◆ If you selected **client**, **server**, or **client and server** for either C++ or Java, you must select a WSDL file, as well as a service and port.
 - ◆ If you selected **container** for either C++ or Java, you must select a WSDL file and service.
7. If you selected anything other than the **Empty project** template, click **Next**.
 8. You now have the option of creating a secure project. See the online help for more details.
 9. Click **Finish**.

Creating Other Project Types

Overview

Artix Designer allows you to create Web services projects based on the following application types:

- CORBA
 - EJB
 - Databases
 - z/OS
-

CORBA projects

You can create a Web services project from CORBA IDL input. Thus, without manually editing a WSDL file, you can produce a configuration that routes communication between a CORBA server and SOAP clients

You can specify the IOR details either by referencing a file, entering the IOR directly, or using a naming service.

See the “Creating a CORBA Web services project” topic in the Artix Designer online help for details.

EJB projects

You can expose an existing J2EE application as a Web service by creating a project based on an EJB session bean.

See the “Creating a Web services project from EJB” topic in the Artix Designer online help for details.

Database projects

In a database Web services project, the WSDL operations are implemented by SQL queries and stored procedures defined in the database.

See [“Exposing a Database as a Web Service” on page 63](#) for details.

z/OS projects

You can create a z/OS project based on an existing z/OS application or a WSDL file.

See the “Working with Artix Designer for z/OS” section of the online help, and the [Artix User’s Guide for z/OS](#) for details.

Importing Projects

Overview

If you are one of a team of developers working on the same project, you will probably want to import existing Artix Designer projects from elsewhere in your file system or from a version control system into your workspace.

Importing a project

To import an Artix Designer project:

1. From the main menu bar, select **File | Import**.
2. In the Import wizard, select **General | Existing Projects into Workspace** and click **Next**.
3. Choose either **Select root directory** or **Select archive file** and click **Browse** to locate the directory or archive file containing the projects.
4. Under Projects select the project or projects that you want to import.
5. Click **Finish** to start the import.

Importing preferences

To ensure that all developers working on the same project use the same preferences you can import preferences from a file stored in a shared location.

To import a preferences file:

1. From the main menu bar, select **File | Import**.
2. In the Import wizard, select **General | Preferences** and click **Next**.
3. Click **Browse** and locate the preferences file on the local file system.
4. Select one of the following:
 - ◆ Import all to accept all of the preferences defined in the file.
 - ◆ Choose specific preferences to import to select only specified preferences defined in the file.
5. Click **Finish**.

Sharing launch configurations

Artix Designer also stores Artix Tools launch configurations in your project folders, thus making them visible to version control systems.

See the online help for details.

Generating WSDL

This chapter explains how to generate WSDL from various sources in Artix Designer.

In this chapter

This chapter discusses the following topics:

Creating an Empty WSDL File	page 34
Importing and linking to WSDL files	page 35
Generating WSDL from Different Sources	page 38
Working with XML Schema Files	page 39
Other WSDL-related Tasks	page 40

Creating an Empty WSDL File

Overview

Artix Designer allows you to create a WSDL file from scratch.

Creating a WSDL file

To create an empty WSDL file:

1. Make sure you are in the Artix perspective.
2. Select your Artix Designer project in the Navigator view.
3. From the **File** menu select **New | WSDL File**. The New File wizard launches.
4. In the WSDL File panel, select the folder where you want to store the WSDL file. This can be the root of your project folder or a folder within the project.
5. Type the name of the file in the **File name** field. The `.wsdl` file extension is added for you.
6. Click **Finish**.

The new WSDL file opens in the Source view of the WSDL Editor. See [“Using the WSDL Editor Wizards” on page 46](#) for details on how to add content to the file.

Importing and linking to WSDL files

Overview

Artix Designer allows you to import a WSDL file into your project, either from elsewhere in your file system or from a URL.

In addition, you can link to a WSDL file without importing the physical file into your project folder.

Importing a WSDL file from the file system

To import a WSDL file from a file system directory into your project:

1. Select your Artix Designer project in the Navigator view.
 2. Select **File | Import**. The Import wizard opens.
 3. Select **File System** and click **Next**.
 4. In the File System panel, click the **Browse** button from which you would like to add the resource.
 5. In the panel on the left click the folder icon to display the files in the panel on the right.
 6. Select one or more files from the right-hand panel.
 7. Specify the project or folder that will be the import destination.
 8. Click **Finish**.
-

Importing a WSDL file from a URL

To import a WSDL file from a URL:

1. Select **File | New | WSDL from URL**.
2. In the WSDL File panel, select the folder where you want to store the WSDL file.
3. In the **File name** field, type the name of the file as you want it to appear in your workspace.
4. In the **URL Path** field enter the full URL for the WSDL file, for example:

```
http://api.google.com/GoogleSearch.wsdl
```

5. Click **Finish**.

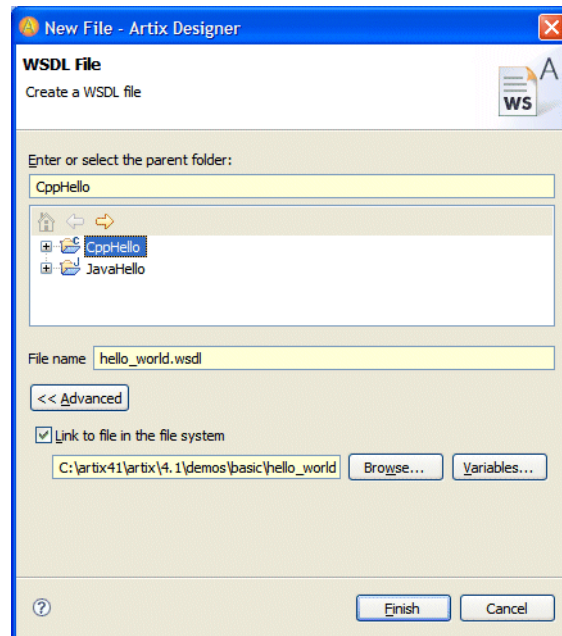
A copy of the remote WSDL file is created in your workspace and the file is opened in the WSDL Editor.

Linking to a WSDL file

To link to a WSDL file:

1. Make sure you are in the Artix perspective.
2. Select your Artix Designer project in the Navigator view.
3. Select **File | New | WSDL File**. The New File wizard launches.
4. In the WSDL File panel, select the folder where you want to store the WSDL file.

Figure 5: *Linking to a WSDL File*



5. In the **File name** field, type the name of the file as you want it to appear in your workspace.
6. Click the **Advanced** button.
7. Select the **Link to file in the file system** checkbox.
8. Click the **Browse** button to locate the file that you want to link to.
9. Click **Finish**.

The WSDL file opens in the WSDL Editor and appears as a link under your project folder in the Navigator view.

Figure 6: *The Linked WSDL File Icon*



Generating WSDL from Different Sources

Overview

With Artix Designer you can generate WSDL from the following sources:

- IDL
 - Fixed and tagged datasets
 - EJB session beans
 - Java classes
-

Generating WSDL from IDL

Artix Designer allows you to expose a CORBA application as a Web service by generating WSDL from CORBA IDL.

Generating WSDL from fixed record length data

You can generate a WSDL file from fixed record length data. Common uses for this type of payload format are communicating with back-end services on mainframes and applications written in COBOL.

Generating WSDL from tagged data

You can use Artix Designer to generate WSDL from documents that are self-describing or where the data fields are delimited by specified characters.

Generating WSDL from an EJB session bean

You can expose an existing J2EE application as a Web service by generating a WSDL file based on an EJB session bean.

You have access to the EJB JAR file containing the session bean before launching the WSDL from EJB wizard.

Generating WSDL from a Java class

Artix Designer enables you to create a WSDL file in which the logical content describes the API of a Java class.

Creating WSDL from Java is a three-step process:

1. Java-enabling your project.
2. Adding the class to the build path.
3. Running the WSDL from Java wizard.

See the “Creating WSDL from Java” online help topic for full details.

Working with XML Schema Files

Overview

You might prefer to use XSD to store a set of XML Schema types, as an alternative to creating a list of types directly in your WSDL files.

Creating an XSD file

To create an empty XSD file:

1. Make sure you are in the Artix perspective.
 2. Select your Artix Designer project in the Navigator view.
 3. Select **File|New|XSD File**. The New File wizard launches.
 4. In the XSD File panel, select the folder where you want to store the schema.
 5. Type the name of the file in the **File name** field. The `.xsd` file extension is added for you.
 6. Click **Finish**.
-

Importing an XSD file from a URL

To import an XSD file from a URL:

1. Select **File|New|XSD from URL**.
2. In the XSD File panel, select the folder where you want to store the XSD file.
3. In the **File name** field, type the name of the file as you want it to appear in your workspace.
4. In the **URL Path** field enter the full URL for the XSD file.
5. Click **Finish**.

A copy of the remote XSD file is created in your workspace and the file is opened in the WSDL Editor.

Importing a file from the file system

To import an XSD file from elsewhere in your file system, follow the same procedure as described in [“Importing a WSDL file from the file system” on page 35](#).

Other WSDL-related Tasks

SOAP-enabling a WSDL file

If you have an Artix contract with a fully defined interface that you plan to expose as a Web service using SOAP over HTTP, you can quickly add a default SOAP binding and a SOAP port to the contract.

You must provide the following details:

- The name of the interface from which to generate the binding
- How you want the SOAP binding configured
- The address of the HTTP port the service will be exposed on
- The names for the binding and the service. Artix Designer does the rest.

To SOAP-enable a WSDL file, select the file and then select **Artix Designer | SOAP Enable**.

CORBA-enabling a WSDL file

If you have an Artix contract with a fully defined interface that you plan to expose as a CORBA service, you can quickly add a a CORBA binding and a CORBA port to the contract.

You must provide the name of the interface from which to generate the binding, a name for the binding, and a name for the service.

To CORBA-enable a WSDL file, select the file and then select **Artix Designer | CORBA Enable**.

You will need to edit the generated `port` element to have a valid IOR.

Example 1: *A CORBA Service Generated by the CORBA Enable Option*

```
<service name="MyCORBAService">
  <port binding="tns:MyCORBABinding" name="MyCORBAPort">
    <corba:address location="..." />
  </port>
</service>
```

Creating an access control list

Artix enables you to create access control lists (ACLs) to determine which roles can access the different operations contained within a WSDL file's port types.

You can define ACLs at the port type level and have them applied to all operations for that port type. Or you can define them at the operation level.

Once created, ACLs for each WSDL file in your Artix project are stored in XML format under a folder labeled *WSDLFileName_ACLS*.

To create an access control list for a WSDL file, select the file and then select **Artix Designer|New Access Control List**.

Using the WSDL Editor

This chapter explains how to edit Artix contracts using the WSDL Editor.

In this chapter

This chapter discusses the following topics:

Introducing the WSDL Editor	page 44
Using the WSDL Editor Wizards	page 46
Setting WSDL Editor Preferences	page 47

Introducing the WSDL Editor

Why use the WSDL Editor?

Artix Designer's WSDL Editor is integrated with its code generation tools, and incorporates a thorough understanding of the Artix extensions to the WSDL standard.

For example, Artix Designer automatically adds the required namespace declarations and prefix definitions when you build Artix applications that involve Artix-extended data marshalling, transport protocols, or routing.

WSDL Editor views

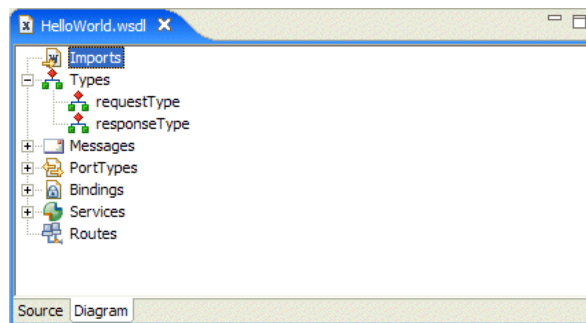
The WSDL Editor is comprised of two views:

- The Diagram view
- The Source view

The Diagram view displays the different WSDL elements in a graphical tree-like hierarchy.

The Source view allows you to view and edit the WSDL in text-editor mode.

Figure 7: *The Diagram View in the WSDL Editor*

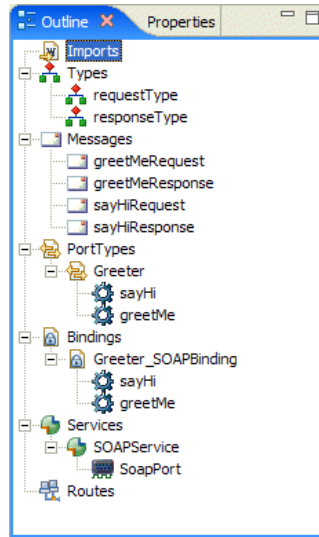


Using the Outline view

You can use the Eclipse Outline view in conjunction with the WSDL Editor Source view.

Each element in the WSDL file is displayed as a node in the Outline view. You can click a node in the Outline view to jump to that element in the WSDL source.

Figure 8: *The Outline View*



Using the WSDL Editor Wizards

Overview

There are a number of wizards in the WSDL Editor that enable you to add, edit, or delete elements in a WSDL contract.

Launching a wizard

The most important wizards are accessible from the Artix Designer menu and toolbar. These allow you to create new:

- Types
- Messages
- PortTypes
- Bindings
- Services
- Routes

You can also create new WSDL elements by right-clicking the appropriate node in the Diagram view of the WSDL Editor.

Right-clicking a node also allows you to edit or delete certain elements.

For more details ...

For detailed help on each of the WSDL Editor wizard, launch a wizard and press **F1** for context-sensitive help.

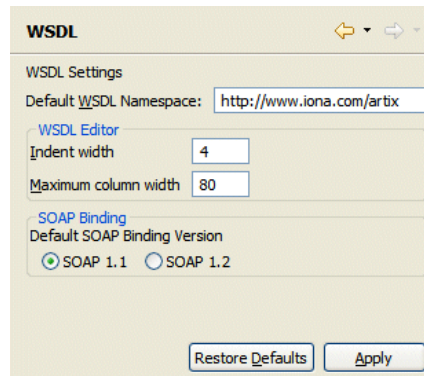
Alternatively, launch the Eclipse Help, and in the Contents pane, select **Artix Designer | Using the WSDL Editor**.

Setting WSDL Editor Preferences

Setting basic preferences

The WSDL Settings page in the Artix Designer Preferences panel allows you to choose a default WSDL namespace for any WSDL file that you create and to set some basic formatting options for the Source view in the WSDL Editor. It also allows you to choose a default version of SOAP when creating SOAP bindings.

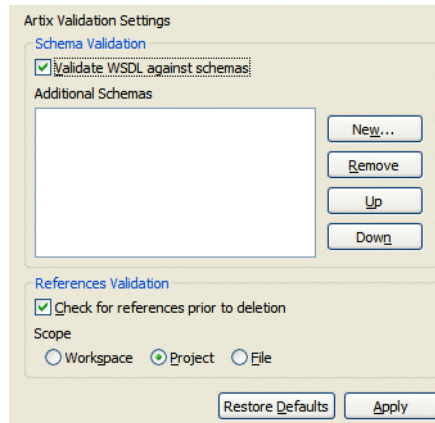
Figure 9: *Artix Designer Preferences—WSDL Settings*



Setting validation settings

To ensure that your WSDL is validated against the XML Schema files that ship with Artix, select the **Validate WSDL against schemas** checkbox.

Figure 10: *Artix Designer Preferences—Validation Settings*



You can also add your own schemas to the Additional Schemas list by clicking **New** and choosing a schema from your file system.

For more details ...

See [“Setting Artix Designer Preferences”](#) on page 20.

Generating, Launching, and Deploying Artix Solutions

This chapter explains how to generate, deploy, launch, and test Artix Web services.

In this chapter

This chapter discusses the following topics:

Introducing Launch Configurations	page 50
Generating Code	page 53
Launching Artix Applications	page 55
Deploying a Service into a Container	page 57
Launching a Container	page 58
Testing a Service	page 60

Introducing Launch Configurations

What is a launch configuration?

In Eclipse, a launch configuration describes how to launch a program.

Using Artix launch configurations

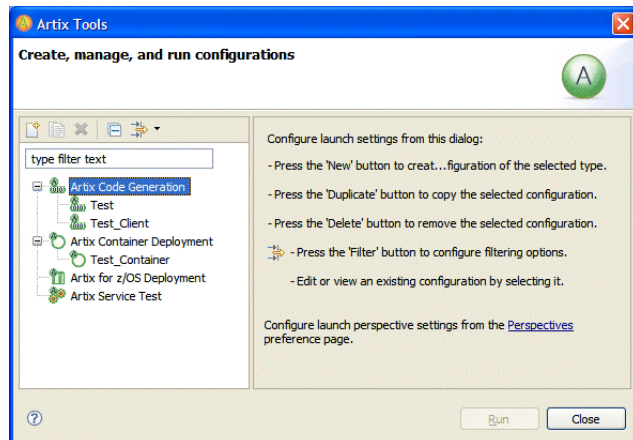
Artix Designer includes the following launch configurations:

- Code Generation—Allows you to generate code based on a WSDL file
- Container Deployment—Enables you to deploy a service plug-in into an Artix container
- z/OS Deployment—Allows you to deploy the files needed to support Web service client requests on mainframe applications. See the [Artix User's Guide for z/OS](#) for more details
- Service Test—Enables you to manually test a deployed service

Opening the Artix Tools window

To access the Artix launch configurations, select **Artix Designer | Artix Tools | Artix Tools** from the menu bar.

Figure 11: *The Artix Tools Window*



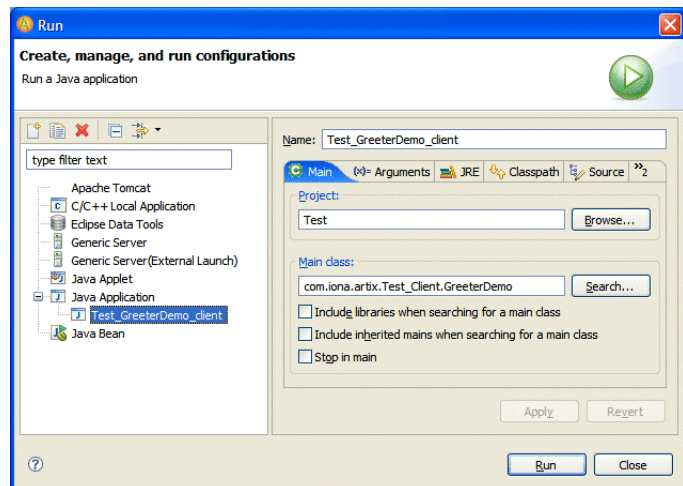
Using Eclipse launch configurations

Once you have used the Artix Tools window to generate your Web service, you can launch the generated application or container using the following Eclipse launch configurations.

- Java application launch configuration
- External program launch configuration

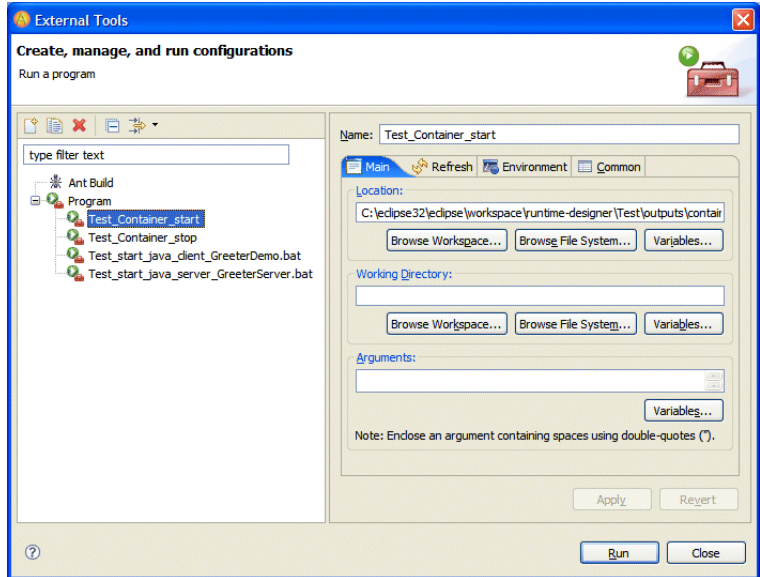
You can launch Java client or server applications by selecting **Run | Run** from the menu bar.

Figure 12: *Launching Java Applications from the Run Window*



You can launch all application types and Artix containers by selecting **Run | External Tools | External Tools**.

Figure 13: *Launching Applications from the External Tools Window*



For more details ...

See the “Generating, launching, and deploying Artix solutions” section of the online help.

Generating Code

Overview

You can use Artix Designer to create code generation launch configurations for the following:

- service plug-ins in both C++ and Java
- applications in C++ and Java
- CORBA IDL
- routers
- types

Creating a code generation configuration

To create a code generation configuration:

1. In the Artix Tools window, select the **Artix Code Generation** node.
2. Click the New launch configuration icon.



3. Enter a name for the code generation configuration in the **Name** text box.
4. Edit the entries in each of the General, Generation, and WSDL Details tabbed pages. For details see the “Generating code” section of the online help.
5. Click **Apply** to save the configuration.
6. Click **Run** to generate the code based on your selections.

Compiling code automatically

Any code that you generate in Artix Designer is compiled for you automatically by the Eclipse JDT compiler or by your C++ compiler, provided that you have selected **Build Automatically** from the **Project** menu. If you are developing C++ code on Windows, you must source your version of Visual C++ in `start_eclipse.bat`, the file that launches Eclipse with the Artix Designer plug-ins enabled.

To edit the `start_eclipse` script:

1. Using a text editor, open the `start_eclipse.bat` file from one of the following locations:

Artix-installed Eclipse

```
ArtixInstallDir/artix/4/0/bin
```

Existing Eclipse

```
EclipseInstallDir
```

2. Add the following line above the call to `artix_env.bat`:

Visual C++ 6.0

```
call "C:\Program Files\Microsoft Visual  
Studio\vc98\bin\vcvars32.bat"
```

Visual C++ 7.1

```
call "C:\Program Files\Microsoft Visual Studio .NET  
2003\Common7\Tools\vsvars32.bat"
```

Launching Artix Applications

Overview

Once you have run your code generation configuration and made any necessary changes to the code, you are ready to run the generated application.

How you launch the application depends on whether you built it in Java or C++.

You can also launch both Java and C++ applications by running the start scripts created by the code generation process.

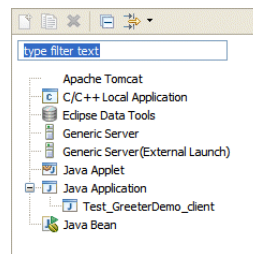
Launching Java applications

Thanks to Artix Designer's close integration with the Eclipse JDT, a Java application launch configuration is created automatically for your application as soon as you generate the Java code.

To run a Java application launch configuration:

1. From the menu bar, select **Run | Run**.
2. In the Run window, expand the Java Application node in the list of configurations on the left.
3. Select the launch configuration for your client or server application.

Figure 14: *Running a Java Application*



4. Select the **Arguments** tab and enter any arguments that our application requires in the **Program Arguments** field.
5. Click **Run**.

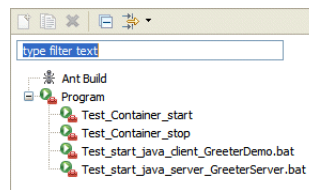
The application process launches in the Eclipse Console view.

Launching C++ applications

To launch a C++ application:

1. From the menu bar, select **Run | External Tools | External Tools**.
2. In the External Tools window, expand the **Program** node in the **Configurations** tree on the left.
3. Select the launch configuration for your client or server application.

Figure 15: *Running a C++ Application*



4. In the Main tabbed page, enter any arguments that your application requires in the **Arguments** field.
5. Click **Run**.

Server applications launch in a new command window. Output for client applications displays in the Console view.

Launching applications from start scripts

When you use a code generation configuration to create an Artix application, start and stop scripts are added to the following location:

```
ProjectName\outputs\applications\CodeGenConfigName\bin
```

You can launch both Java and C++ applications by running the appropriate start script from the command prompt.

Note: If an application takes any arguments, you need to edit its start script accordingly.

Deploying a Service into a Container

Overview

You can use the Artix Tools window to deploy a service plug-in into an Artix container.

You can deploy a service that you have created yourself by running a code generation configuration—[“Generating Code” on page 53](#).

Alternatively, you can deploy the following standard Artix services into a container:

- Locator
- Session manager
- Login service

Creating a container deployment configuration

To create a container deployment configuration:

1. In the Artix Tools window, select the **Artix Container Deployment** launch configuration.
2. Click the New launch configuration icon.



3. Enter a name for the container deployment configuration.
4. Edit the entries in the General, Container Basics, and Advanced QoS Options tabbed pages:
5. Click **Apply** and then **Run**.

What happens next depends on the deployment option that you selected in the General page.

If you chose to deploy the container, you are prompted to switch to the Container Admin perspective. See [“Managing Services and Containers” on page 71](#) for details.

If you chose to generate deployment scripts only, you remain in the Artix perspective. The scripts are written to the following directory:

```
ArtixProjectName/outputs/containers/ContainerConfigName
```

See [“Launching a Container” on page 58](#) for details of how to launch the container from the generated scripts.

Launching a Container

Overview

Once you have created a container in Artix Designer, you can launch it in one of the following ways:

- From a start script
 - From the Artix Tools window
-

Launching from a start script

When you generate a container, either from the Artix Tools window or by using a project template, a start script is added to your Artix Designer workspace.

You can run the start script by either:

- Double-clicking it in the Navigator view
- Launching it from the External Tools window

To run a start script from the External Tools window:

1. From the menu bar, select **Run | External Tools | External Tools**.
2. In the External Tools window, expand the **Programs** node in the **Configurations** tree on the left.
3. Select the `ServiceName_container_start` script and click **Run**.

Note: When you launch a container from a start script, the container does not appear in the Artix Container Admin perspective automatically. To add it to the perspective, check the port that the container is running on by opening the start script in a text editor and then add the container to the Admin perspective manually.

Launching from the Artix Tools window

The advantage of launching a container from the Artix Tools window is that the container appears in the Artix Container Admin perspective automatically.

When you create a container deployment launch configuration in the Artix Tools window, you can choose to start the container and deploy the service plug-in by selecting either the **Deploy only** or **Deploy and generate deployment scripts** options in the General tabbed page and then clicking **Run**.

When you create a container by using a project template, the container deployment scripts are created for you, but the container is not started and the service is not deployed. You can run the container from the Artix Tools window by selecting the **Deploy using previously generated scripts** option in the General tabbed page.

Testing a Service

Overview

You can use Artix Designer to test your Web services.

The service test launch configuration creates a dynamic client that you can use to test any running Web service, whether it was created in Artix or not. The service can be deployed as a server application or as a service plug-in.

The service test launch configuration supports all Artix bindings and transports and an extensive number of types.

Creating a service test configuration

To create a service test launch configuration:

1. In the Artix Tools window, select the **Artix Service Test** node and then click **New**.
 2. Enter a name for the service test configuration.
 3. Edit the entries in the General, Service, Operation, and Expected Response tabbed pages.
 4. Click **Apply** and then **Run**.
-

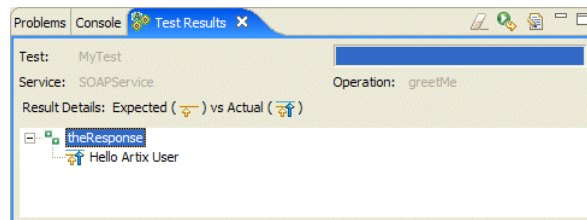
Using the Test Results view

When you run a service test, the results are displayed in the Test Results view.

The contents of the view depends on whether you provided an expected response in your service test configuration.

If you did not provide an expected response, it displays the actual results of the test.

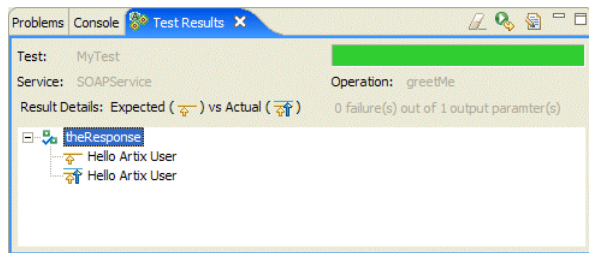
Figure 16: *The Test Results View with an Actual Result Displayed*



If you provided an expected response, the expected and actual results are compared.

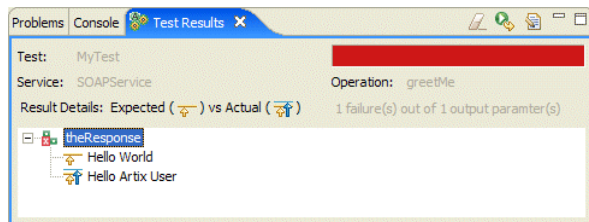
A green bar shows that the test was successful.

Figure 17: *Actual and Expected Results where the Test was Successful*



And a red bar shows that the test failed.

Figure 18: *Actual and Expected Results where the Test Failed*



Exposing a Database as a Web Service

Artix Designer allows you to expose a database as a Web service.

In a database Web services project, the WSDL operations are implemented by SQL queries and stored procedures defined in the database.

In this chapter

Exposing a database as a Web service in Artix involves the following steps:

Creating a Database Web Services Project	page 64
Creating a Database Web Services Project	page 64
Adding Operations to the DB Config File	page 65
Generating and running the Web service	page 68

Creating a Database Web Services Project

Creating the project

To create a database Web services project:

1. From the **File** menu select **New | Project**.
2. In the New Project dialog box, expand the **IONA Artix Designer** folder.
3. Click **Database Web Services Project** and then click **Next**.

The Database Web Services Project wizard opens.

You are asked to provide a name and destination for your database project, as you would for any Artix Designer project.

Creating a bookmark

A database bookmark is a key concept in the database project. A bookmark describes the connection to your database.

The Database Web Services Project wizard creates a bookmark for you. You will have to:

- Provide the location of the JDBC driver for your database management system
 - Provide connection details for your database
 - Enter a name for the bookmark
-

For more details ...

See the “Working with database projects” section of the online help.

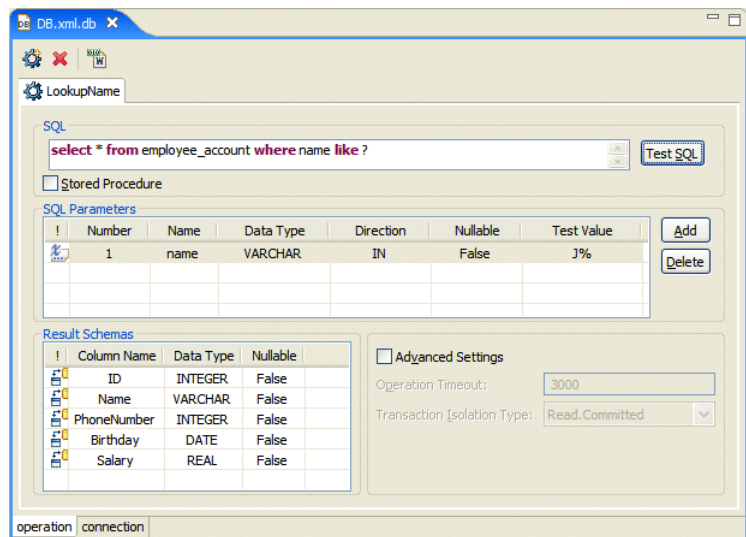
Adding Operations to the DB Config File

What is a DB config file?

When you create a database project, an empty database configuration file is added to your workspace. This DB config file is the source from which you will generate your WSDL file and code by mapping WSDL operations to SQL queries and stored database procedures.

The name of the DB config file takes the form *ProjectName.xml.db*.

Figure 19: *The DB Config Editor*



Adding operations

To add an operation to a DB config file:

1. In the Operation0 tab, enter the SQL statement or stored procedure on which you want to base your operation in the **SQL** field.

Note: The SQL statements or stored procedures must conform to the SQL-92 standard and JDBC 1.x.

For example:

SQL statement

```
SELECT * FROM employee_account WHERE ID=?
```

Stored procedure with parameters

```
{Call add_salary_inout(?,?,?)}
```

Stored procedure without parameters

```
{Call get_all}
```

2. If what you have entered in the SQL field is the name of a stored procedure, select the **Stored Procedure** checkbox.
3. If your SQL query contains parameters, you must provide details for each parameter in the **SQL Parameters** grid.

Figure 20: Adding Parameters to a Stored Procedure Operation

The screenshot shows a configuration window for a web service operation. The 'SQL' field contains the query: `{Call add_salary_inout(?,?,?)}`. The 'Stored Procedure' checkbox is checked. Below this is the 'SQL Parameters' table, which lists three parameters: 'iden' (INTEGER, IN, False, 1), 'na' (VARCHAR, OUT, False, Artix), and 'sa' (INTEGER, INOUT, False, 1). There are 'Add' and 'Delete' buttons next to the table. Below the parameters table is the 'Result Schemas' table, which lists columns: 'ID' (INTEGER, False), 'Name' (VARCHAR, False), 'PhoneNumber' (INTEGER, False), 'Birthday' (DATE, False), and 'Salary' (REAL, False). To the right of the result schemas is the 'Advanced Settings' section, which includes 'Operation Timeout' set to 3000 and 'Transaction Isolation Type' set to 'Read, Committed'.

!	Number	Name	Data Type	Direction	Nullable	Test Value
	1	iden	INTEGER	IN	False	1
	2	na	VARCHAR	OUT	False	Artix
	3	sa	INTEGER	INOUT	False	1

!	Column Name	Data Type	Nullable
	ID	INTEGER	False
	Name	VARCHAR	False
	PhoneNumber	INTEGER	False
	Birthday	DATE	False
	Salary	REAL	False

- ◆ Click **Add** and edit the default values for the parameter.
- ◆ The value in the **Number** cell corresponds to the question mark in the SQL query. You cannot change the numbers in this field. To change the order of parameters, delete parameters as necessary.

- ◆ Select a JDBC data type for the parameter from the **Data Type** drop-down list.
 - ◆ If you are using a stored procedure, select a direction for the parameter from the **Direction** drop-down list.
 - ◆ If the parameter can handle a null value, select **True** from the **Nullable** drop-down list.
 - ◆ Enter a test value in the **Test Value** cell.
4. Click the **Test SQL** button. The query will be written to the DB config file and the results will appear in the SQL Results view.
-

Renaming an operation

To give the operation a more meaningful name:

1. In the Outline view, expand the **Operations** node.
2. Right-click the **Operation0** node.
3. Select **Rename** from the pop-up menu.

Alternatively, right click the **Operation0** tab and select **Rename** from the pop-up menu.

Adding operations

To add another operation, click the **Add Operation** button and repeat the steps above.

Once you have added all the operations required for your Web service, you are ready to generate the service.

Generating and running the Web service

Generating the service

Once you have added operations to the DB config file, you are ready to generate the Web service.

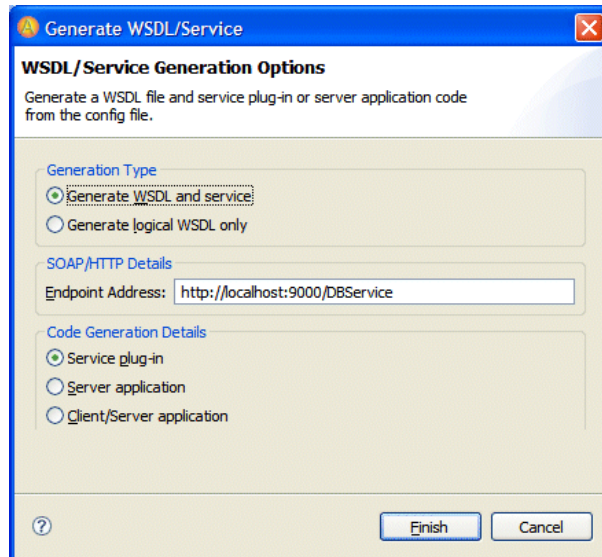
Generating a Web service performs the following steps:

- Creates a WSDL file based on the values entered in the DB config file
- Generates the code for a Java service plug-in, server application, or client/server application
- Uses Eclipse's JDT to compile the generated code and creates the related launch configurations

To generate a service from a DB config file, click the **Generate WSDL/Service** button at the top of the DB Config Editor.

The Generate WSDL/Service wizard opens.

Figure 21: *The Generate WSDL/Service Wizard*



Here you can choose whether to generate logical WSDL only or generate a complete WSDL file and one of the following:

- A service plug-in
 - A server application
 - Both a server and a client application
-

Running the Web service

Once you have successfully generated your database Web service, you can run the generated server application or deploy the service plug-in into a container.

If you chose to deploy your Web service as a server application, you can run it as you would any Artix Java application. See [“Launching Artix Applications” on page 55](#) for more details.

Similarly, if you chose to also generate a client/server application you can run the client against the server in the same way. Remember to include any required parameters when running the client.

If you chose to deploy your Web service as a service plug-in, you need to deploy the service by creating a container deployment launch configuration. See [“Deploying a Service into a Container” on page 57](#).

You can then generate a client based on the same WSDL that you generated from the DB config file and run it against the service.

Managing Services and Containers

This chapter explains how to manage Artix services and containers using the Container Admin perspective.

In this chapter

This chapter discusses the following topics:

Introducing the Container Admin perspective	page 72
Container Admin Tasks	page 75

Introducing the Container Admin perspective

Overview

The Container Admin perspective allows you to view and manage any number of Artix service plug-ins running in one or more Artix containers.

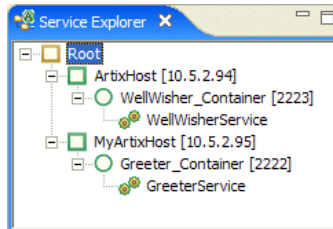
The perspective consists of the following views:

- Service Explorer
- Services List
- Console Events
- Properties

The Service Explorer view

The Service Explorer provides a navigator tree that allows you to see all the services to which you are currently connected, arranged according to the containers and host computers on which they are running.

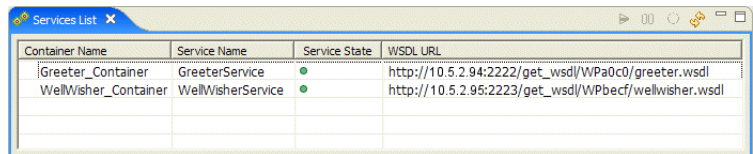
Figure 22: *The Service Explorer View*



The Services List view

The Services List displays all the services for the currently selected node in the Service Explorer tree. This view displays the service's current status, the name of the container on which it is running, and the path to the WSDL file for the service.

Figure 23: *The Services List View*



Container Name	Service Name	Service State	WSDL URL
Greeter_Container	GreeterService	●	http://10.5.2.94:2222/get_wsd/WPa0c0/greeter.wsdl
WellWisher_Container	WellWisherService	●	http://10.5.2.95:2223/get_wsd/WPbecf/wellwisher.wsdl

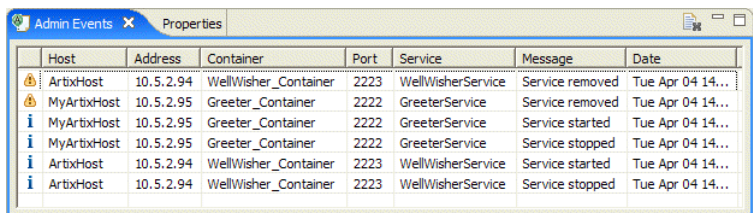
You can use the icons at the top right of the view to start, stop, and remove a service, and to refresh a container.

The Admin Events view

The Admin Events view records the events that have occurred within the Container Admin perspective.

You can reorder these events according to the column headings at the top of the view by clicking on a heading. And you can clear the log by right-clicking anywhere in the view and selecting **Clear Admin Events** from the pop-up menu.

Figure 24: *The Admin Events View*



Host	Address	Container	Port	Service	Message	Date
ArtixHost	10.5.2.94	WellWisher_Container	2223	WellWisherService	Service removed	Tue Apr 04 14...
MyArtixHost	10.5.2.95	Greeter_Container	2222	GreeterService	Service removed	Tue Apr 04 14...
MyArtixHost	10.5.2.95	Greeter_Container	2222	GreeterService	Service started	Tue Apr 04 14...
MyArtixHost	10.5.2.95	Greeter_Container	2222	GreeterService	Service stopped	Tue Apr 04 14...
ArtixHost	10.5.2.94	WellWisher_Container	2223	WellWisherService	Service started	Tue Apr 04 14...
ArtixHost	10.5.2.94	WellWisher_Container	2223	WellWisherService	Service stopped	Tue Apr 04 14...

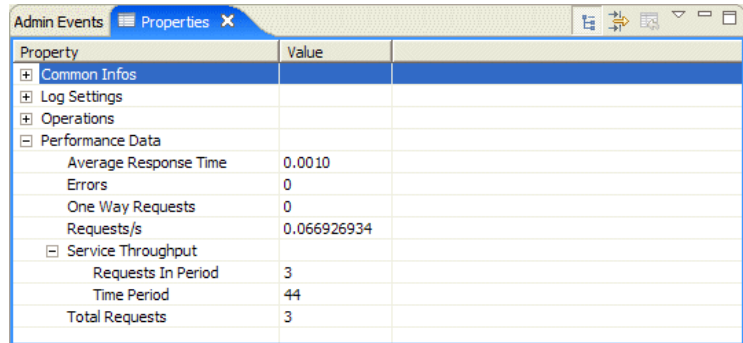
You can choose to have admin events saved to a file by selecting the appropriate option in the Container Admin page of the Artix Designer preferences.

The Properties view

The Properties view provides the following functions:

- Displays properties for each node that you select in the Service Explorer view, for example, host, container, and service.
- Allows you to set logging severity levels for the Artix bus and selected subsystems.
- Displays performance data for the services running in the container admin perspective.

Figure 25: *The Properties view*



Property	Value
Common Infos	
Log Settings	
Operations	
Performance Data	
Average Response Time	0.0010
Errors	0
One Way Requests	0
Requests/s	0.066926934
Service Throughput	
Requests In Period	3
Time Period	44
Total Requests	3

Opening the Container Admin perspective

To open the Container Admin perspective, select **Window | Open Perspective | Artix Container Admin** from the menu bar.

Note: You can choose to open the Container Admin perspective automatically when you create a container deployment configuration in the Artix Tools window.

Container Admin Tasks

Overview

You can manage each node in the Container Admin perspective by right-clicking it and selecting an option from the pop-up menu.

Managing containers

Once you have started an Artix container, either from the command line, or from within Artix Designer, you can add it to the console.

You can then deploy services into the container and monitor their performance using the Properties view.

Using the default container

The default container is a container that you can choose to start automatically every time the Container Admin perspective is run.

This allows you to quickly add and test services without going outside the console to start a container.

You enable the default container in the Container Admin page of the Artix Designer preferences.

Managing services

You can use the Container Admin perspective to:

- Deploy services into a container
 - Stop and start services
 - Remove services from a container
-

Monitoring performance

You can use the Properties view to monitor the performance of services running in the Container Admin perspective.

Among the performance data recorded in the Properties view are:

- Average response time
 - Service throughput
 - Total number of requests
-

For more details ...

For a full description of what you can do in the Container Admin perspective, see the “Managing Services and Containers” section of the Artix Designer online help.

Securing Artix Applications

This chapter explains how to enable the security features available in Artix using Artix Designer.

In this chapter

This chapter discusses the following topics:

Introducing Artix Security	page 78
Creating secure projects	page 79

Introducing Artix Security

Overview

When you create a Web services project in Artix Designer, you have the option of creating:

- An insecure project
 - A secure project with the Artix security disabled
 - A secure project with Artix security enabled
-

Secure projects with Artix Security disabled

If you create a secure project, but with Artix security disabled, the following security features will be available to your Artix application:

- Encryption using X.509 certificates
 - Authentication performed during SSL/TLS handshake at the TCP/IP socket layer
-

Secure projects with Artix Security enabled

You also have the option of enabling Artix security, which is a comprehensive security framework that supports authentication and authorization using data stored in a central Artix Security Service.

Artix Designer enables you to employ the following security layers in your Artix applications:

- Transport Layer Security (TLS)—Implemented at the TCP/IP socket layer, TLS provides for encryption, message integrity, and authentication using X.509 certificates.
- Web Services Security (WSS)—WSS implements security at the SOAP layer. This security mechanism is used to transmit WSS username and password credentials.
- Common Secure Interoperability (CSI)—CSI allows for username and password authentication when using a CORBA binding.
- Authorization (server side)—The server program supports authorization using the Artix security framework. A server-side plug-in calls out to a central security service in order to figure out which operations the client is authorized to invoke.

Creating secure projects

Overview

You can enable security in any of the basic web services project templates, apart from the empty project template.

Creating a secure project

To enable security for your Artix Designer project:

1. Create a basic web services project. See [“Creating Basic Projects” on page 29](#).
2. Select any template *except* the **Empty Project** template.
3. In the Security panel of the wizard, select **Secure Communication** in the Security panel of the wizard.

Customizing security settings

Once you have created a basic web services project with security enabled, you will need to customize the project's security settings, to ensure that your application is truly secure.

You can also use this method to make a previously insecure application secure, and vice-versa.

To customize your project's security configuration:

1. Open the Artix Tools window. See [“Generating Code” on page 53](#).
2. Select the launch configuration for your application or container from the list of configurations on the left.

3. Update the settings in the Security tabbed page. See the online help for details.

Figure 26: *The Security Tabbed Page in the Artix Tools Window*

General | Generation | WSDL Details | WS-RM | **Security** | Common

Transport Communication
 Insecure Secure

Security Service Config
 Artix Security Service Enable access control for Web service
 Host: localhost
 TLS Port: 53112
 Security Service Trusted CA: C:/artix41/artix/4.1/demos/security/certificate: Browse...
 Security Service Server Certificate: C:/artix41/artix/4.1/demos/security/certificate: Browse...
 Security Service Server Password File: C:/artix41/artix/4.1/demos/security/certificate: Browse...

Security Fields
 Server Security Fields | Client Security Fields | Username Password Credentials | Router Security Fields
 Server Trusted CA: C:/artix41/artix/4.1/demos/security/certificates/tls/x509/trusted_ca_lists/ca_list:
 Server Certificate: C:/artix41/artix/4.1/demos/security/certificates/tls/x509/certs/services/administr
 Server Password File: C:/artix41/artix/4.1/demos/security/certificates/tls/x509/certs/services/administr

Generation output folder: /secure/secure

4. Click **Run** to regenerate your Artix configuration file and any other security-related scripts and files.

Index

A

- access control lists 40
- applications
 - generating 53
 - launching 55
- Artix demos 25
- Artix Designer
 - features 16
 - installing 17
 - launching 18
 - perspectives 21
 - preferences 20, 32, 47
 - projects 27–32
 - supported platforms 17
- Artix Tools window 50
- authentication 78
- authorization 78

C

- C++
 - generating 53
 - launching 55
- cheat sheets 23
- code
 - generating 53
- containers
 - deploying 57
 - deploying services into 75
 - launching 58
 - managing 75
- CORBA 31, 40

D

- databases 21, 31
 - projects 63
- demos 25

E

- Eclipse 16
- EJB 31, 38
- encryption 78

F

- fixed datasets 38

H

- help 23

I

- IDL
 - generating 53
 - generating WSDL from 38
- installing 17

J

- Java
 - generating 53
 - generating WSDL from 38
 - launching 55

L

- launch configurations 32, 50
- launching
 - applications 55
 - Artix Designer 18
 - containers 58

P

- perspectives 21
- preferences 20, 32, 47
- projects 27–32

S

- security 77–80
- services
 - deploying 57
 - generating 53
 - managing 75
 - testing 60
- SOAP 40, 47
- SQL 63, 66
- starting 18
- stored procedures 63, 66

INDEX

supported platforms 17

T

tagged datasets 38

testing 60

W

WSDL Editor 43–48

WSDL files

 CORBA-enabling 40

 creating access control lists for 40

 creating empty 34

 editing 43–48

 generating 38

 importing from a URL 35

 linking to 36

 SOAP-enabling 40

X

XML Schema 39

XSD 39

Z

z/OS 21, 31, 50