Get.It!
Quick Start Guide
Release 2.0.1
September 2001

ITEM GET-2.0.1-ENG-01001-00088

Peregrine Systems, Inc.
3611 Valley Centre Drive
San Diego, CA 92130
www.peregrine.com

This edition applies to version 2.0.1 of the licensed program

Get.It!

Contents

Introduction

Get.It! Overview **1-1**

About this Guide **1-2**

Organization of this Guide 1-3

Conventions Used in this Guide 1-4

Installing and Configuring Get.It!

Installation Requirements **2-2**

Software 2-2

Windows 2-2

Solaris 2-2

Linux 2-2

Hardware 2-3

Pre-installation Considerations **2-3**

Installing Get.It!—Windows **2-4**

Installing an Additional Weblication **2-25**

Installing Get.It!—Solaris or Linux with JRun **2-28**

Building the Files on Windows 2-28

Configuring the Installation on Solaris or Linux 2-29

Configuring the AssetCenter amdb.ini File 2-35

Using the Admin Module to Configure Get.It! 2-36

Loading the ServiceCenter Files 2-38

Configuring Get.It! to use the Demo Weblication 2-41

Server Processing Considerations — Multiple JVMs 2-41

Adding a JRun Server 2-42
Adding Multiple JVM Connectors 2-45
Updating the JVM Script Path 2-46
Configuring Multiple Virtual Directories 2-46
Updating the archway.ini File 2-47
Setting Heap Memory for Single or Multiple JVMs 2-48

Setting Parameters for the Production Environment 2-51

Internationalization 2-52

Setting the Content Type Encoding 2-52
Setting the Locale Code 2-53
Setting the Currency Code 2-56

Configuring the Application Servers

Initial Application Server Configuration 3-1

Tomcat 3-2
WebLogic 3-3
WebSphere 3-4
Reinstalling Get.It! on WebSphere 3-12

Updating the scriptpoller.ini Files 3-14

Testing the Installation 3-14

Get.It! Architecture Overview

Get.It! Architecture 4-1

Archway Internal Architecture 4-3

Archway Requests 4-4
The Document Manager 4-7

Get.It! Administration

Using the Admin Module 5-1

Accessing the Admin Module 5-1
Using the Control Panel 5-3
Viewing the Server Log 5-4
Using the Settings Page 5-6
Setting Parameters using the Admin Module 5-7
Setting Parameters in the archway.ini File 5-14
Verifying Script Status 5-15
Displaying Message Queues 5-15
Showing Queue Status 5-15
Displaying Form Information 5-16
Displaying Form Details 5-18

Security

Get.It! Security Overview 6-1

User Registration 6-2

User Authentication 6-3

Overriding the Login Script 6-4

Windows NT Challenge/Response 6-6

Setting Up Windows NT Challenge/Response 6-7
Updating the loginverify.asp File 6-8
Setting Permissions for the Presentation Folder 6-8
Testing the Settings 6-9
Setting the Default Login as login.asp 6-10

Adapters

Identifying the Back-end System Version 7-2

ServiceCenter Adapter 7-3

Parameters 7-3
ServiceCenter Event Handling 7-4
Using the _event parameter 7-4

Troubleshooting the ServiceCenter Database Connection 7-5

AssetCenter Adapter 7-6

Get.It! INI Parameters 7-6
Troubleshooting the AssetCenter Database Connection 7-6
AssetCenter Feature Links 7-7

JDBC Adapter 7-8

Adding a JDBC Adapter 7-8
Verifying the System DSN 7-9
Updating the archway.ini File 7-12
Validating the JDBC Adapter Connection 7-13
Adding the New Database Settings 7-15
Creating a New Weblication 7-16
Preventing Lost Connections 7-16
Calling a Stored Procedure 7-17
Using a URL 7-18
Using a JScript 7-18
Using a Document Insert Or Update 7-18

LDAP Adapter 7-20

Connecting LDAP to Get.It! 7-20
Updating the archway.ini File 7-20

E-mail Adapter 7-21

Verifying Adapter Connections 7-24

Troubleshooting

Verifying Installation and Configuration 8-1

AssetCenter ODBC Error 8-1
Configuring JRun to Interact with the Desktop 8-1
Creating a System DSN 8-2

JRun Issues 8-3

Cannot Connect to JRun 8-3
Reinstalling the Connector Wizard 8-3
JRun Session Cookie 8-6

Log Files **8-6**

AssetCenter Adapter Error Codes **8-7**

Chapter 1
Introduction

The *Get.It! Quick Start Guide* includes instructions for installing and configuring Peregrine Systems® Get.It!. The guide also describes the administrative steps you can take to monitor the server connections, perform user administration, change control settings, configure adapters, and set up Windows NT Challenge/Response.

Get.It! Overview

Get.It! includes:

- Archway—a Java servlet that processes HTTP requests from a browser, sends them through an adapter to a back-end system, and returns XML data to be displayed in the browser.
- Adapters—SCAdapter, ACAdapter, JDBC adapter, LDAP adapter, E-mail adapter.
- Demo weblication—a sample weblication provided as an example for learning how to use Get.It! Studio to edit and build your own weblications. Use the JDBC adapter to connect to the sample database included.
- Get.It! Studio—a comprehensive tailoring utility that allows you to edit weblications or create your own. Get.It! Studio is also used to build software updates provided by Peregrine.
- ServiceCenter® files—to allow you to use ServiceCenter as your back-end system.

Get.It! architecture has been designed to use a Web interface that can:

- Offer services to everyone in an organization.
- Offer access everywhere users need it.
- Offer support related to everything in the infrastructure that helps employees get things done.

The Get.It! architecture is designed to make services available to users through common interfaces like Web browsers, handheld computers (PalmPilot), or mobile phones. The applications are designed to provide a wide range of services, from helping a user with a PC problem, to allowing the creation of a purchase request, to reporting a problem with the employee's office space.

Get.It! is the base architecture for many Peregrine products. It can be used to connect to ServiceCenter and AssetCenter® through special adapters provided with the Get.It! installation. Peregrine weblications—Get.Service! and Get.Resources!—can be purchased separately to interface with these systems.

About this Guide

The *Get.It! Quick Start Guide* is intended for use by an administrator or developer who will be installing or maintaining a Get.It! system. This guide should be used in conjunction with several other manuals, which are:

- Operating guides, reference manuals, and other documentation for your PC hardware and operating software.
- The *Get.It! Studio Introduction* and the *Get.It! Tailoring Guide* which describe how to use Get.It! Studio to customize Get.It! weblications.
- The Quick Start Guides for the Peregrine weblications you have purchased.
- JRun documentation located online in the ...JRun\docs directory.

To use this manual effectively, you should have a working knowledge of both the PC hardware and operating system, and of the database management for the back-end systems you are linking to Get.It! (such as Peregrine's ServiceCenter and AssetCenter).

Note: Refer to the ServiceCenter and AssetCenter manuals for instructions on installing and using these systems.

Organization of this Guide

The following table shows you which parts of the guide you need to look in to find the information you need.

To Find This	Look Here
Background information; how to use this guide.	Chapter 1, "Introduction"
Procedures for installing Get.It!, configuring the connection to AssetCenter and ServiceCenter.	Chapter 2, "Installing and Configuring Get.It!"
How to configure the alternate application servers.	Chapter 3, "Configuring the Application Servers"
Information about how Get.It! works.	Chapter 4, "Get.It! Architecture Overview"
Using the Admin module; resetting the server link to the back-end systems; monitoring the server log.	Chapter 5, "Get.It! Administration"
Get.It! security; user authentication; setting up Windows NT Challenge/Response.	Chapter 6, "Security"
Detailed information about each of the adapters provided with Get.It!.	Chapter 7, "Adapters"
Verifying the connections to AssetCenter and ServiceCenter; troubleshooting the JRun configuration; using log files.	Chapter 8, "Troubleshooting"

Conventions Used in this Guide

Screen shots in this guide are included as examples only. In some cases, the Peregrine products AssetCenter and ServiceCenter have been used for examples of screens and to create sample XML or ECMAScript code.

Screens are shown using the Get.It! Classic stylesheet.

The following documentation conventions are used in this guide:

Object	Font	Example
Button	New Century Schoolbook Bold	Click **Next**
File name	Arial Bold	The **login.asp** file

Sample script or XML code	Courier New	`var msgTicket = new Message( "Problem" );`
		
		`...`
		`msgTicket.set( "_event", "epmc" );`
		The ellipsis (`...`) is used to indicate that portions of a script have been omitted because they are not needed for the current topic. Samples of code are not entire files, but they are representative of the information being discussed in a particular section.

Book title	New Century Schoolbook Italic	Refer to the *Get.It! Quick Start Guide*.
Menu option	New Century Schoolbook Bold	Select **Start>Program Files**.

Chapter 2
Installing and Configuring Get.It!

This chapter leads you through the installation and configuration of Get.It! 2.0.1 on three platforms:

- Windows NT Server and Windows 2000 Server
- Solaris
- Linux

During the installation, you will have the opportunity to install the following:

- Java Run-time Environment (JRE) 1.3.1, International version.
- JRun 3.1, a Java application server.
- Get.It!, including Get.It! Studio with a demo weblication, and the Peregrine weblications you have purchased—Get.Resources!, Get.Service!, and Get.Answers!.

Instructions for adding and configuring multiple Java virtual machines (JVMs) are included. Multiple JVMs are supported with JRun only.

Get.It! 2.0.1 also includes support for the following application servers:

- WebSphere 4.0
- WebLogic 6.0 SP1
- Tomcat 3.2.x

Instructions for configuring these application servers are included. Refer to the individual application server documentation for installation instructions.

Note: If you are going to use Get.It! to connect to ServiceCenter or AssetCenter, you will need to install these products separately. The back-end systems are *not* included on the Get.It! CD. The installation instructions for ServiceCenter and AssetCenter can be found in their respective system manuals.

Installation Requirements

This section outlines the recommended minimum configuration for proper installation and configuration of Get.It!. Before beginning the installation of Get.It!, ensure that you have the following.

Software

Windows

- Operating system: Microsoft Windows NT Server, version 4.0 SP4 or later, or Windows 2000 Server.
- Web server: Microsoft IIS Server 4.0 or 5.0 (available from the Microsoft Web site).
- If you will be using Get.It! with AssetCenter, you must have version 3.02 or later installed. Get.It! 2.0.1 also supports AssetCenter 3.5 SP2 and later, 3.51, and 3.6.

  The AssetCenter API must be installed on the same system as Get.It!; however, the AssetCenter database can be on another system. This means that when you install AssetCenter, you must either proceed with: (1) a full installation, or (2) a custom installation, selecting the "AssetCenter API" option. AssetCenter must be installed and have a valid connection established to the database on the Get.It! server. Get.It! uses the **amdb.ini** file to determine how to attach to the AssetCenter database.
- If you will be using Get.It! with Peregrine's ServiceCenter, you must have ServiceCenter 3.0 SP3 or later, or ServiceCenter 4.x installed.
- If you are installing Get.It! in Japanese, you must install on Japanese Windows.

Solaris

- Operating system: Solaris 2.7 or 2.8
- Web server: Apache 1.3.20 or Netscape Enterprise Server 4.1 SP5
- Run-time environment: If you will be using Tomcat as your application server, you will need JDK 1.3.1 instead of the JRE 1.3.1 supplied with the Get.It! installation.

Linux

- Operating system: Red Hat 6.2 or 7.1
- Web server: Apache 1.3.20

- Run-time environment: If you will be using Tomcat as your application server, you will need JDK 1.3.1 instead of the JRE 1.3.1 supplied with the Get.It! installation.

Hardware

- Pentium, 400 MHz or faster, with at least 512 MB of RAM. Heap memory should be set to 512 MB. Refer to "Setting Heap Memory for Single or Multiple JVMs" on page 2-48.
- Approximate disk space - 100 MB for Get.It! and 100 MB for Get.It! Studio.

Pre-installation Considerations

Before beginning your installation, review the following information:

- Do you have a Java Run-time Environment installed? Java 2 Runtime Environment 1.3.1 is included on the installation CD. If you are using a version earlier than 1.3.1, install the version on the CD before you begin the Get.It! installation.
- Do you have a supported Java application server installed? JRun 3.1 is provided on the installation CD. If you are using JRun for your application server and have a version earlier than 3.1, uninstall it before you begin the Get.It! installation and install the version on the CD.

  If you will be using one of the alternate application servers, instructions for configuring the supported servers are included in the installation instructions.
- What is your IIS “scripts” directory? The default is usually “C:\Inetpub\scripts.” You will need this information for the JRun Connector Wizard.
- If you do not want to install Get.It! into the default folders, decide where you would like the files to be installed.

Installing Get.It!—Windows

The following installation procedures are for systems using JRun as the application server on a Windows system.

The complete Get.It! installation includes three parts, which must be completed in the following order:

- Install the Java 2 Runtime Environment (JRE).
- Install and connect JRun 3.1 to your Web server.
- Install Get.It!.

> **Important:** The installation paths for JRun and Get.It! must be kept short in order to ensure that all of the Peregrine and supporting Java libraries are correctly loaded by your Java server. **JRun, and Get.It! must be installed at the root level.** If you use long path names during the installation, Get.It! pages may not display.

To install the Java Run-time Environment:

Before you begin the installation, make sure you have completed all of the items listed in "Pre-installation Considerations" on page 2-3.

1. Insert the Get.It! CD into your CD-ROM drive. If the setup does not automatically begin, use the Run command from the Windows Start menu. Run **AutoRun.exe** from the CD ROM drive.

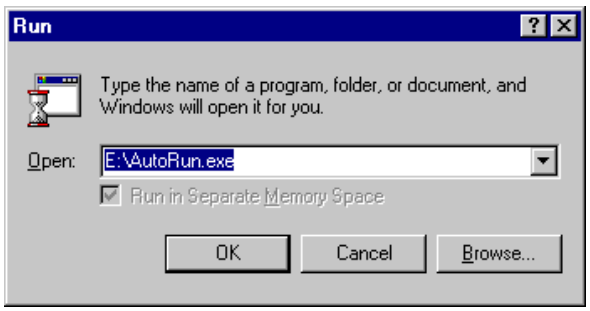

Fig. 2.1 The installation command

The CD Browser is displayed.

Fig. 2.2 CD Browser

2. Click **Install Java 2 Runtime Engine**.

   The Java splash screen is displayed, followed by the license agreement.

3. Click **Yes** to accept the license agreement.

   The Choose Destination Location dialog box is displayed.

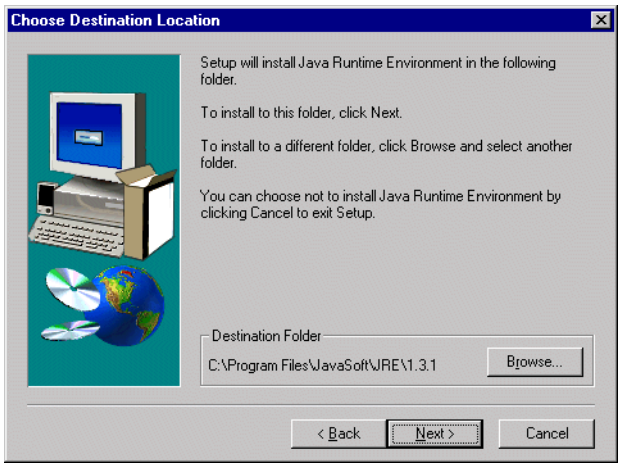

Fig. 2.3 Choose the destination location

4. To install to the default location, click **Next**. Or click **Browse** to edit the path and install to a different location, and then click **Next**.

   Several messages are displayed as the files are installed and the Java JAR files are created.

A message is displayed that the installation is complete. The CD Browser is displayed.

Go to the next procedure to continue the Get.It! installation.

To install JRun 3.1:

1. In the CD Browser, click **Install JRun 3.1**.

Fig. 2.4 CD Browser

The JRun splash screen is displayed, followed by the JRun setup welcome dialog box.

2. Click **Next**.

   The License dialog box is displayed.

3. Click **Yes** to accept the license agreement.

   A dialog box is displayed which asks for a serial number. Leave this field blank. The Get.It! installer will automatically install your JRun license.

4. Click **Next**.

   The Installation Folder dialog box is displayed.

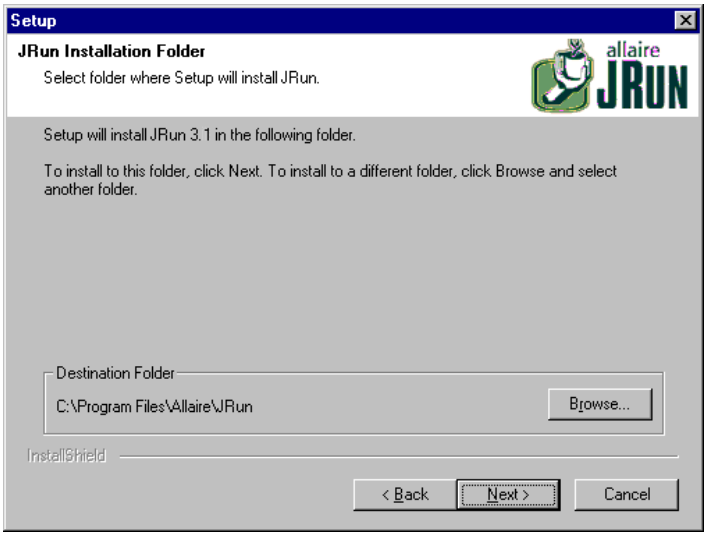

Fig. 2.5 Choose the JRun installation folder

5. Click **Browse** to edit the default location.

Important:	JRun must be installed at the root level of the drive where you are installing JRun (for example, C:\JRun).

6. Click **Next**.

   The Setup Type dialog box is displayed.

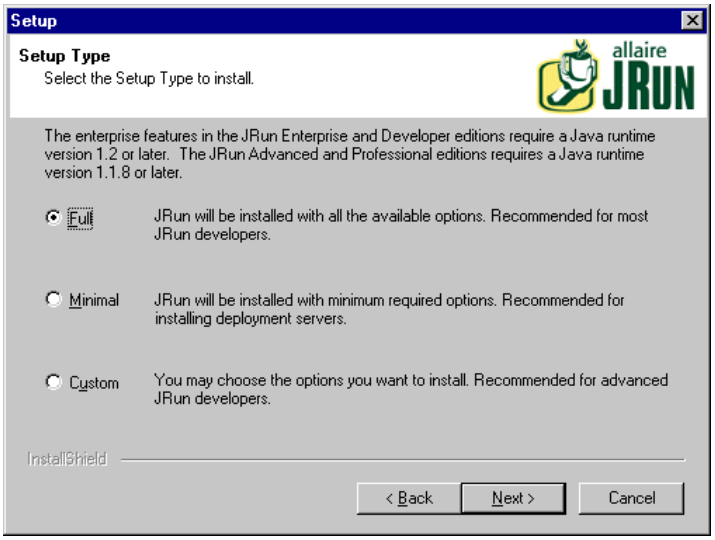

Fig. 2.6 Select the setup type

7. Verify that **Full** is selected, and then click **Next**.

   The Select Program Folder dialog box is displayed.

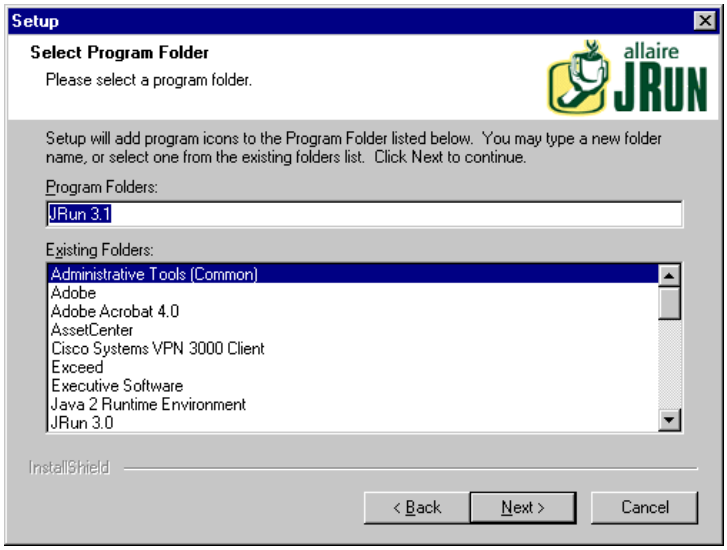

Fig. 2.7 Select the program folder

8. Click **Next**.

   The system will now copy the files to the designated location.

When the process is complete, the Install JRun Services dialog box is displayed.

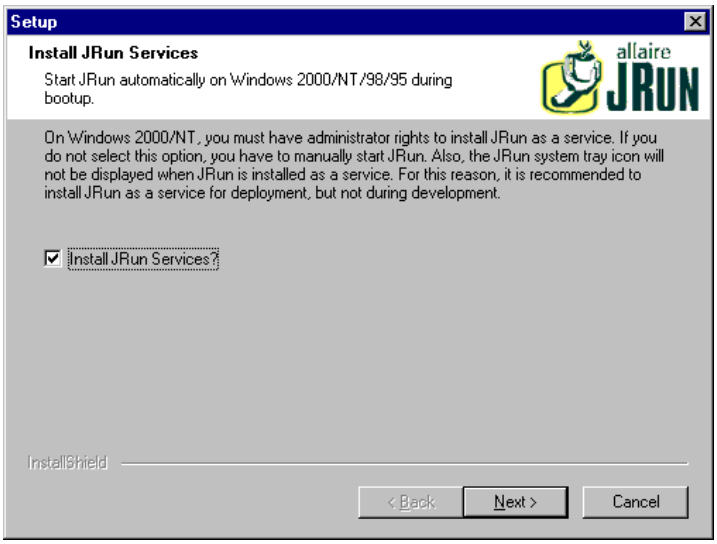

Fig. 2.8 Install JRun Services

9. Verify that the check box is selected, and then click **Next**.

   A dialog box is displayed in which you select the JRE you are going to be using.

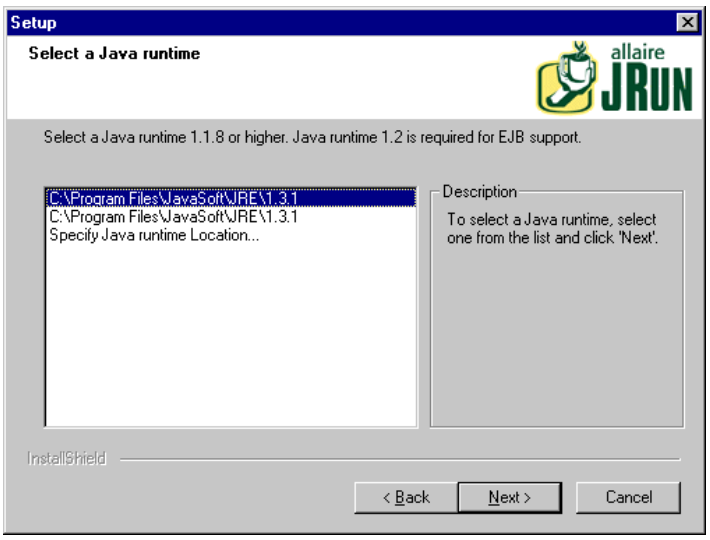

Fig. 2.9 Select the Java RTE

10. Verify that the JRE you installed in the previous procedure is selected, and then click **Next**.

    The JVM Advisor dialog box is displayed. This contains information about settings that are required for Java virtual machine (JVM) configuration.

11. Click **Next**.

    The JRun Management Console dialog box is displayed. This defines the port by which you will access the JRun Management Console. The default is 8000.

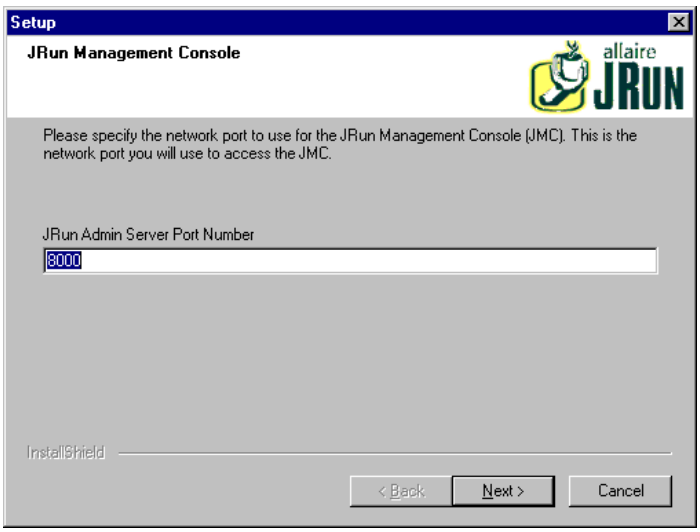

Fig. 2.10 JRun Management Console port number

12. Click **Next** to accept the default port.

    A dialog box is displayed in which you set up a password for the JRun Management Console.

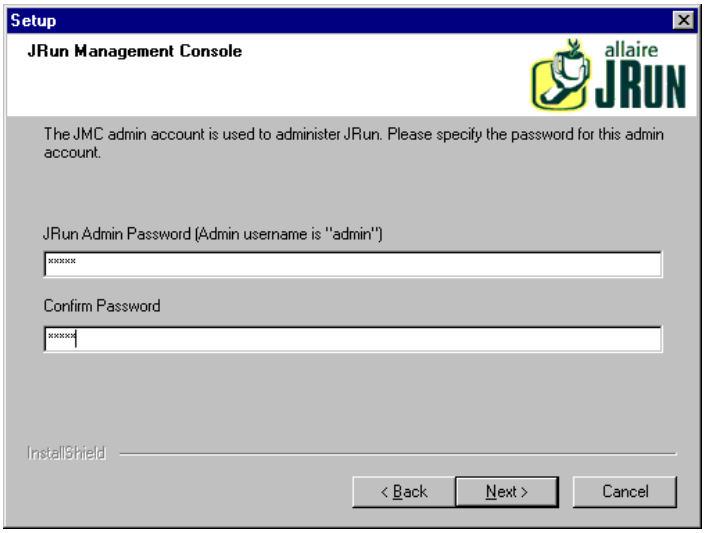

Fig. 2.11 JRun Management Console password

13. Type a password and confirm the password. Click **Next**. Make note of the password—you will need it later in the installation process.

    Several prompts are displayed, followed by a Product Information dialog box.

14. If you would prefer not to receive updates from Allaire, clear the check boxes. Otherwise, leave them selected, and then click **Next**.

    A dialog box is displayed asking if you want to configure JRun to connect to your Web server now or configure the connection at a later time.

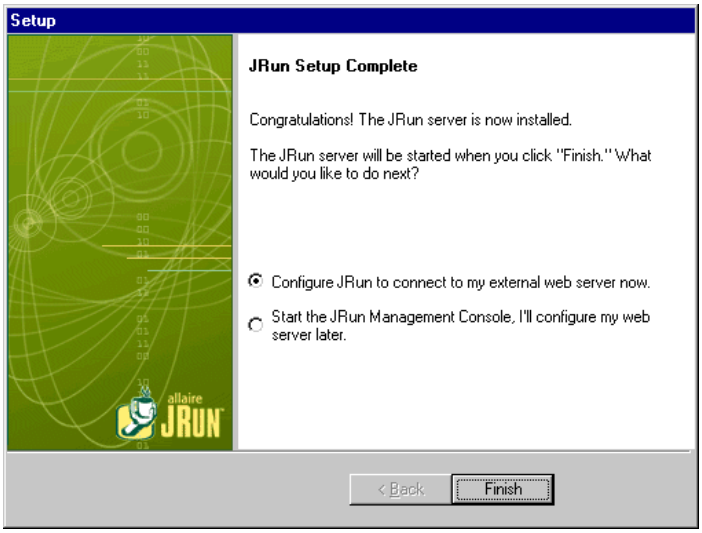

Fig. 2.12 JRun Setup Complete

15. Select to configure JRun now, and then click **Finish**.

    Prompts are displayed indicating that the JRun Admin and Default Servers are being started.

    Your Web browser is launched and the JRun Management Console login screen is displayed.

Fig. 2.13 JRun Management Console login screen

Go to the next procedure to continue your installation of Get.It!.

To configure JRun using the Connector Wizard:

The following steps will configure your JRun connection to the Web server.

1. Log in to the Management Console using a user name of “admin” and the password you chose during the JRun installation. Click **login**.
2. Step 1 in the Connector Wizard: In the JRun Server Name field, select JRun Default Server. In the Web Server Type field, select your Web server from the drop-down box (Personal Web Server for Windows NT Workstation, IIS for Windows NT Server).

   If you select IIS, you will also need to select the version.

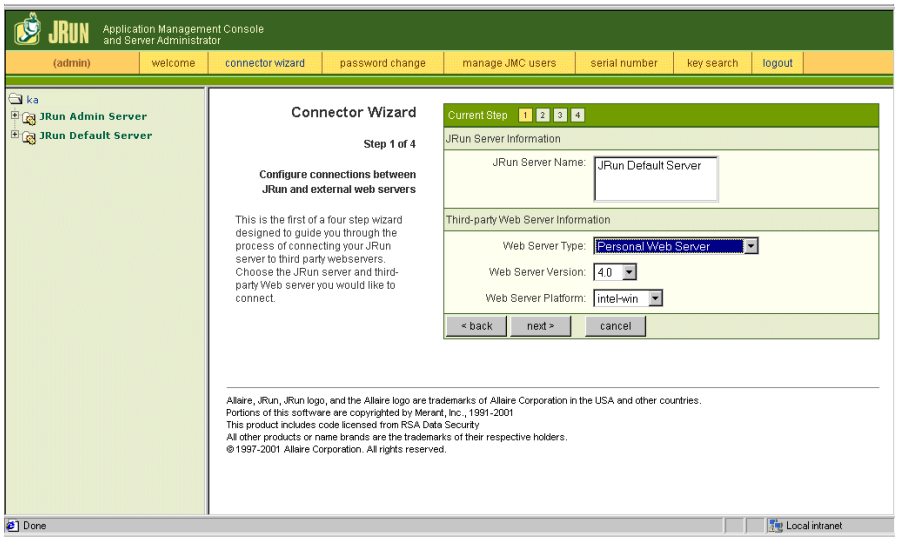

Fig. 2.14 JRun Connector Wizard—Step 1

3. Click **next**.

   If you have not previously stopped your Web server, a prompt is displayed, indicating that you must stop the Web server before proceeding. Click **OK**, and then do the following:

   - Go to **Start>Settings>Control Panel** and double-click the Services icon.
   - Stop the IIS Admin Service. A window is displayed asking if you want to stop the WWW Publishing Service and any other IIS services you have running. Click **OK** to stop these services as well.

4. Step 2 in the Connector Wizard: A default port number of 51000 is filled in. You can use the default number. If you choose to change it, ports which are used by JRun and should not be used here include:

   - 8000—Admin Server
   - 8100—Default Server

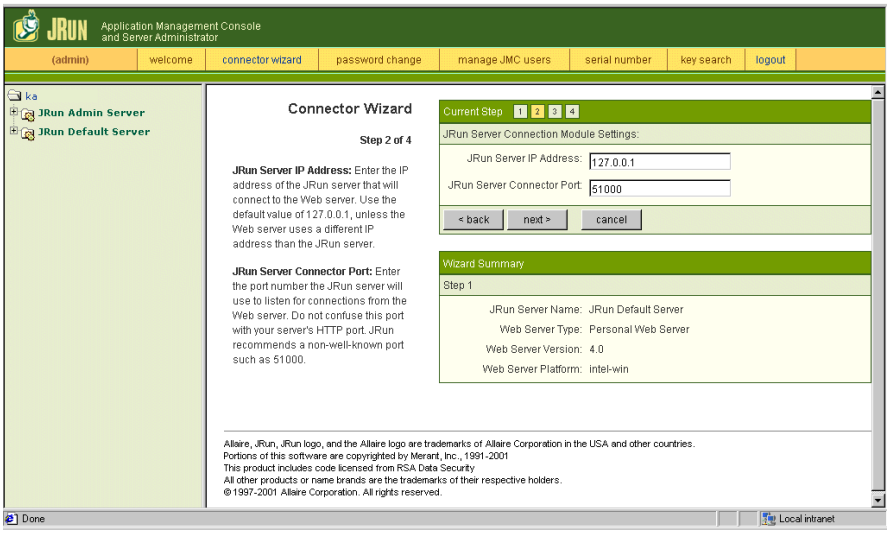

Fig. 2.15 JRun Connector Wizard—Step 2

5. Click **next**.
6. Step 3 in the Connector Wizard: In the PWS or IIS Scripts field, enter the path to the directory where the JRun Connection Module will be installed, or click **Browse** to select a directory. The default is C:\Inetpub\scripts\.

   Verify that the option “Install as a Global Filter” is checked.

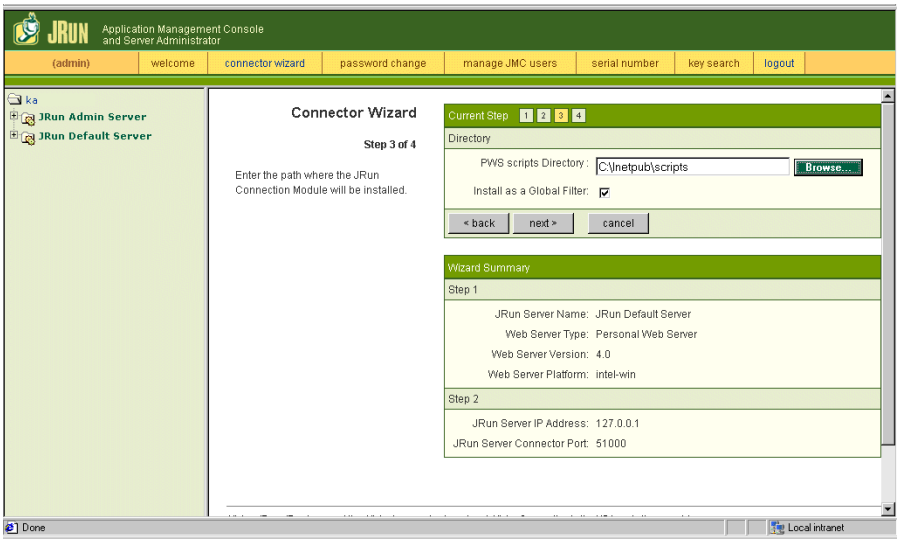

Fig. 2.16 JRun Connector Wizard—Step 3

7. Click **next**.
8. Step 4 in the Connector Wizard: A message is displayed that the connector for the Web server has been successfully installed. Click **done**.

   The main JRun Management Console window is displayed.

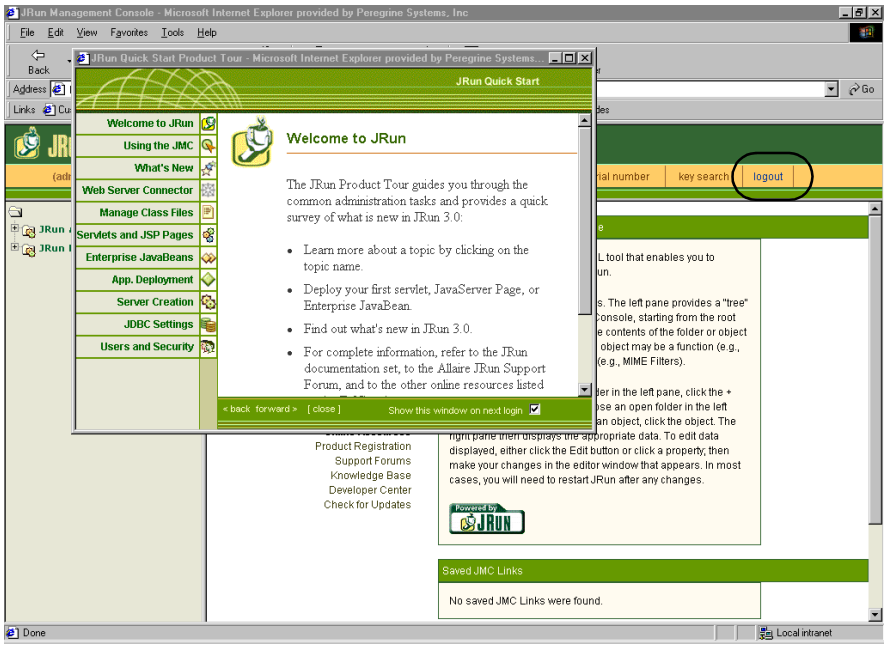

Fig. 2.17 JRun Management Console main window

9. Close the Quick Start Product Tour, and then click **logout**. Close the Web browser.

   The CD Browser is displayed.

To complete the installation, you must install Get.It! and copy the appropriate weblications. Get.It! Studio will automatically create and compile a new project for your installation with your chosen weblications.

Continue to the next procedure to complete your installation of Get.It!.

To install Get.It!:

1. In the CD Browser, click **Install Get.It!**.

Fig. 2.18 CD Browser

The InstallShield Wizard welcome screen is displayed.

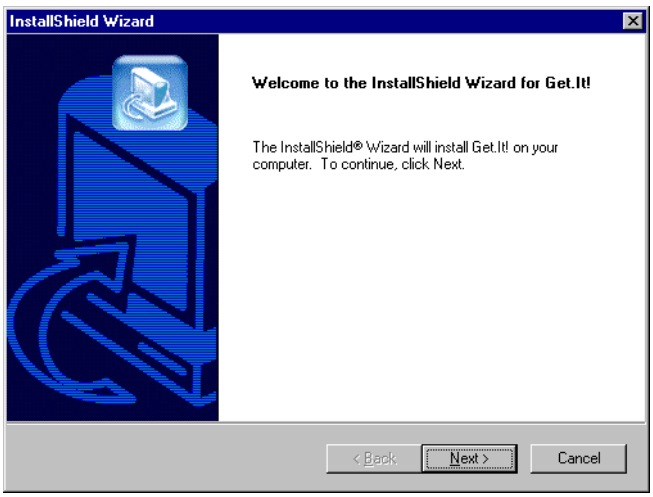

Fig. 2.19 InstallShield Wizard

2. Click **Next**.

   The Destination Location dialog box is displayed. The default location is C:\getit.

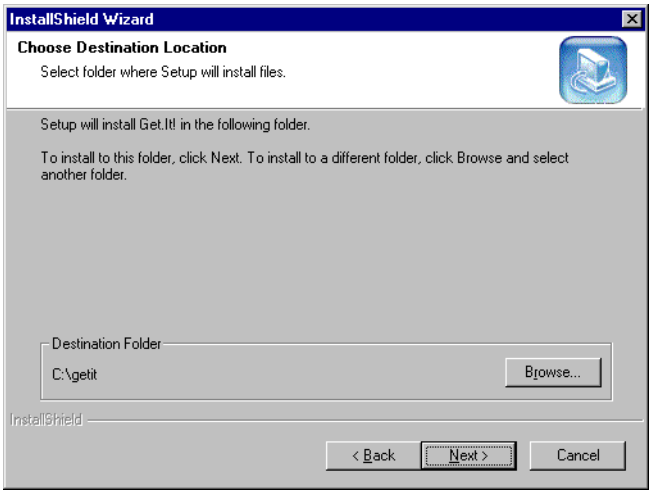

Fig. 2.20 Choose the destination location

3. Click Next to accept the default location. Or click **Browse** to select another location, and then click **Next**.

> **Important:** Get.It! must be installed at the root level (for example, C:\getit).

If you are reinstalling Get.It! and have already installed and configured JRun, click **No** at the prompt and go to step 5.

A message is displayed that JRun and the WWW Publishing Service are being stopped. The Get.It! files are then installed. This will take a few minutes.

A prompt is displayed that the files were successfully installed. A prompt is also displayed that asks if you want to configure JRun.

4. Click **Yes**.

   JRun is configured to use Get.It! and your JRun license is automatically installed.

   A prompt is displayed asking if you would like the installer to create a virtual directory.

5. If you are using IIS, it is strongly recommended that you let the installer create the virtual directory for you. Click **Yes**.

   If you are using another Web server, click **No**. The virtual directory will not be created. After the installation process is complete, create a virtual directory on your Web server that points to the getit\presentation directory.

Prompts are displayed as JRun, IIS, and the World Wide Publishing Service are started.

A prompt is displayed indicating that setup will launch a Wizard to install the weblications.

6. Click **OK**.

   The Get.It! Weblication Installer is displayed.

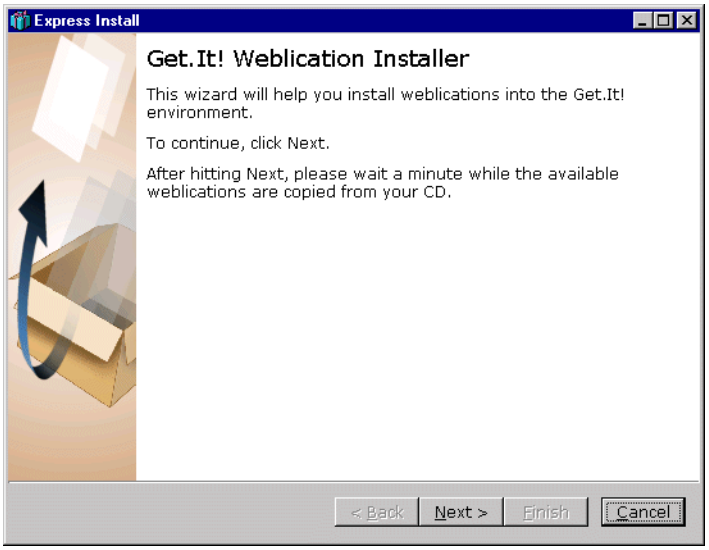

Fig. 2.21 Get.It! Weblication Installer

7. Click **Next**. The weblications available on your CD will be copied. This will take a few seconds.

   When the weblications have been copied, the Choose Weblications dialog box is displayed.

   A demo weblication is included in the list of weblications available. Select to install the demo weblication if you will be doing tailoring using the *Get.It! Tailoring Guide*. The demo weblication is available in English only.

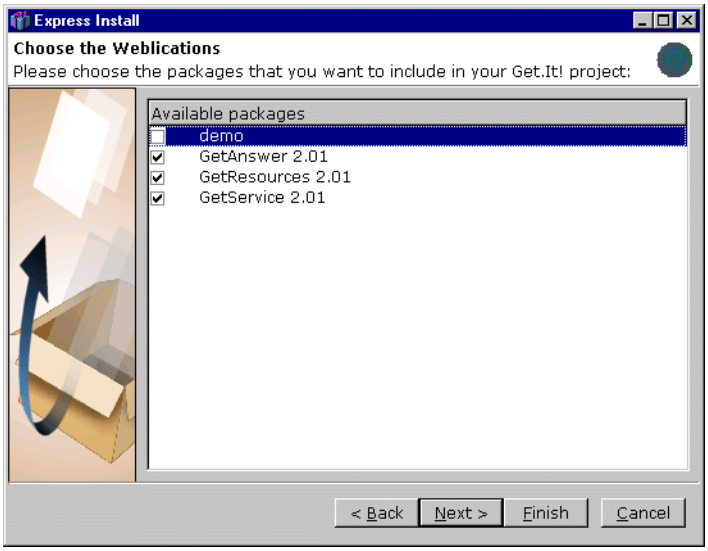

Fig. 2.22 Choose the weblications to install

8. Verify that the weblications you want to install are selected, and then click **Next**.

   A dialog box is displayed in which you select the language you want Studio to use to build your weblications.

Fig. 2.23 Locale selection

If you will be using English, French, German, Italian, or Spanish for your locale, use the default selection.

If you will be using Japanese or Polish, these languages require special builds.

9. Make your selection, and then click **Next**.

   A dialog box is displayed indicating that the project is ready to be built.

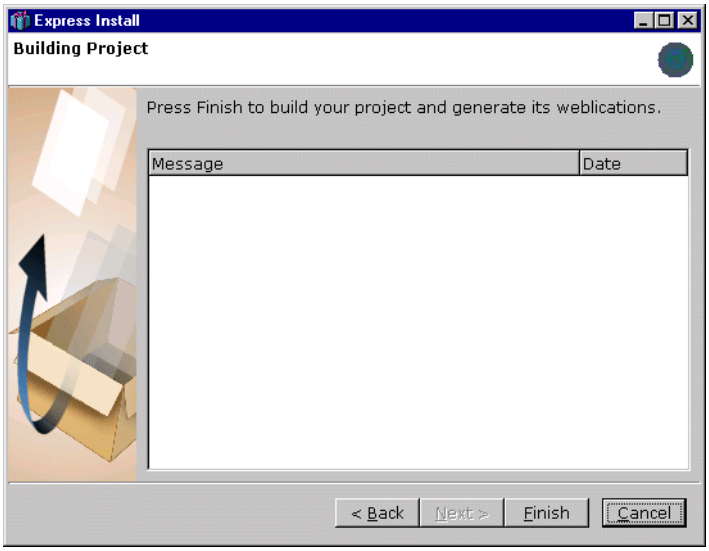

Fig. 2.24 Project ready to build

10. Click **Finish**.

    Confirmation messages are displayed in the dialog box as the projects are built. This process will take several minutes.

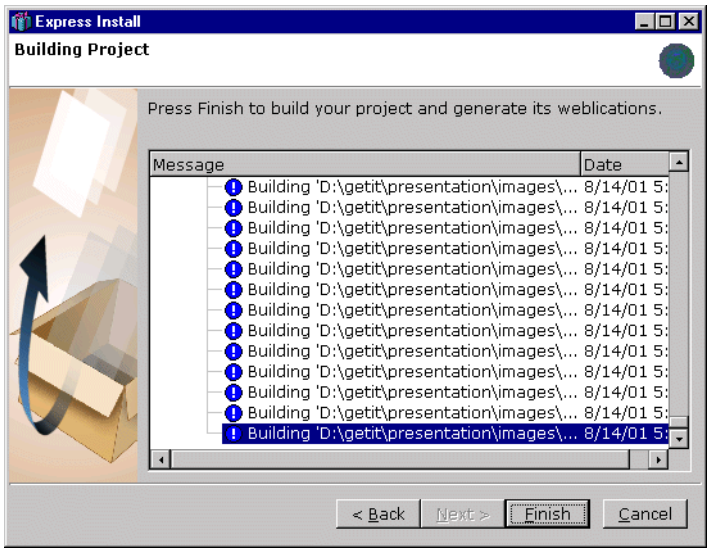

Fig. 2.25 Building the project

When the process is complete, a message is displayed at the bottom of the dialog box that the compilation succeeded.

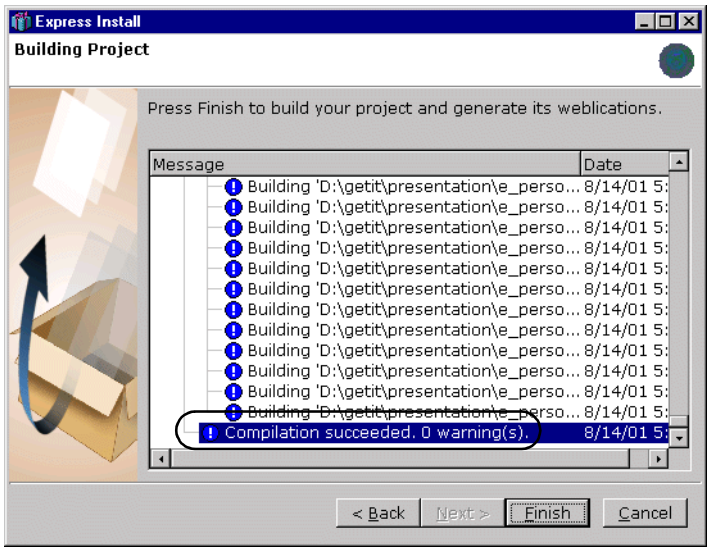

Fig. 2.26 Compilation completed

11. Click **Finish**.

    Prompts are displayed briefly as the installation process is completed.

A dialog box is displayed indicating that the installation is complete. The system will now launch your Web browser and display the Get.It! Administration page. This will enable you to define the Admin settings for the back-end system you are using.

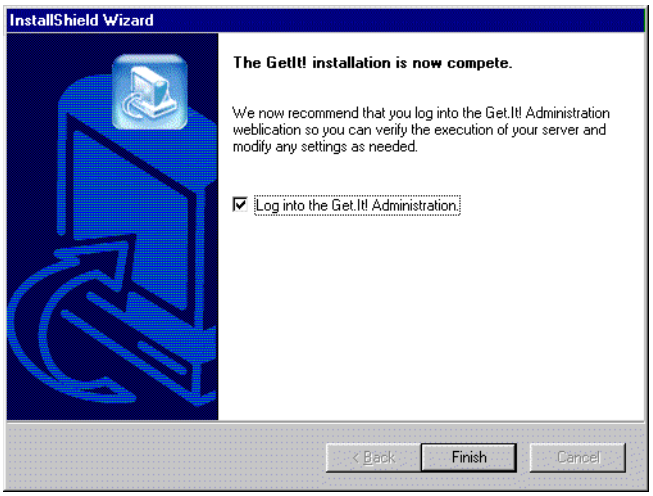

Fig. 2.27 Get.It! Installation complete

12. Click **Finish**.

    Your Web browser is launched and the administration login page (**admin.jsp**) is displayed.

The Get.It! installation is now complete.

The sections that follow include descriptions of the following:

- Using the Admin module to configure Get.It!.
- Loading the ServiceCenter files.
- Configuring Get.It! to use the demo weblication.
- Adding multiple JVMs.
- Setting parameters for the production environment.
- Configuring languages.

Proceed through these sections to fully configure Get.It!.

If you have installed Get.Answers!, refer to the *Get.Answers! Quick Start Guide* for further installation and configuration instructions for this product.

Installing an Additional Weblication

After your initial installation of Get.It!, you may want to install an additional weblication. There are two ways to do this:

- If you have a Get.It! Studio license, the weblication can be installed directly from Studio. Refer to the Get.It! Studio documentation for instructions.
- If you do not have a Get.It! Studio license, you can access the Weblication Installer from the Start menu. This procedure cannot be used for Studio projects that have been customized.

To access the Weblication Installer:

1. Insert the CD for the product you want to install.

Note: If you are installing from the Get.It! CD and the CD autostarts, click cancel. You will be using the CD only for the installer to locate and copy the weblication files.

2. From the Start menu, choose **Programs>Peregrine Get.It!> Weblication Install**.

   The Get.It! Studio splash screen is displayed, followed by the Weblication Installer.

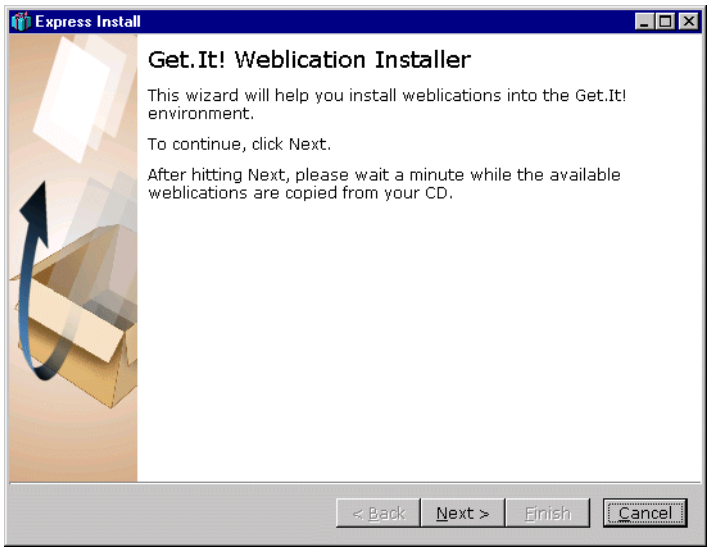

Fig. 2.28 Get.It! Weblication Installer

3. Click **Next**. The weblications available on your CD will be copied. This will take a few seconds.

When the weblications have been copied, the Choose Weblications dialog box is displayed.

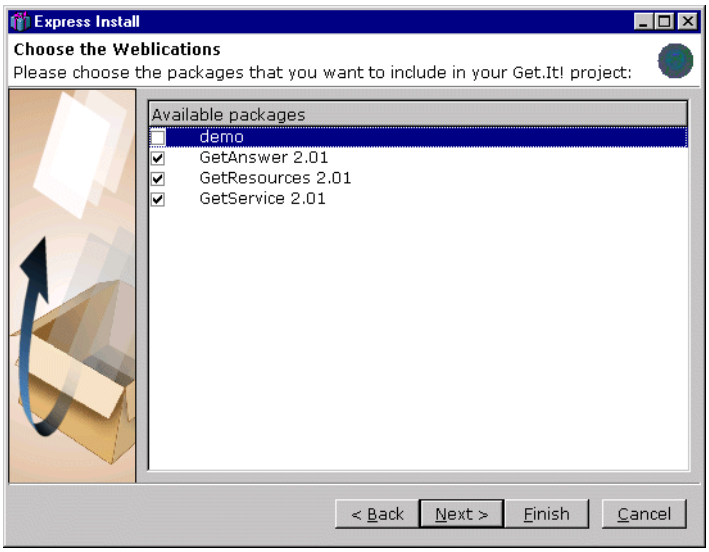

Fig. 2.29 Choose the weblications to install

4. Verify that the weblications you want to install are selected, and then click **Next**. You must select all the packages you want in the new build, including weblications you have already installed.

   A dialog box is displayed indicating that the project is ready to be built.

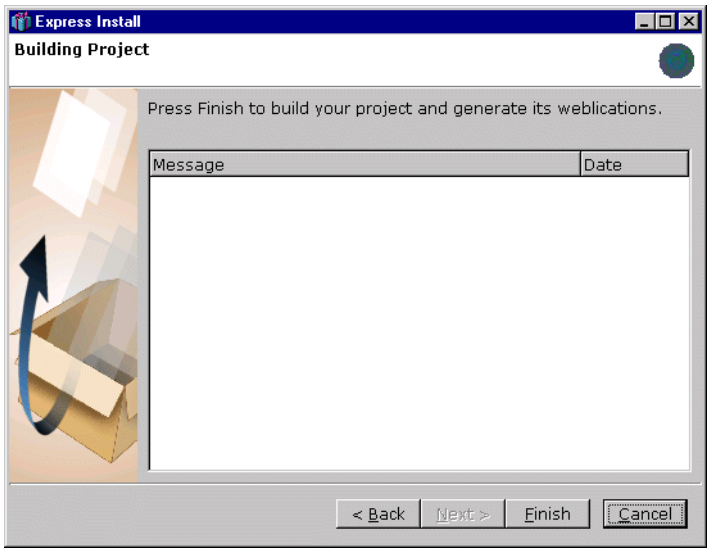

Fig. 2.30 Project ready to build

5. Click **Finish**.

   Confirmation messages are displayed in the dialog box as the projects are built. This process will take several minutes.

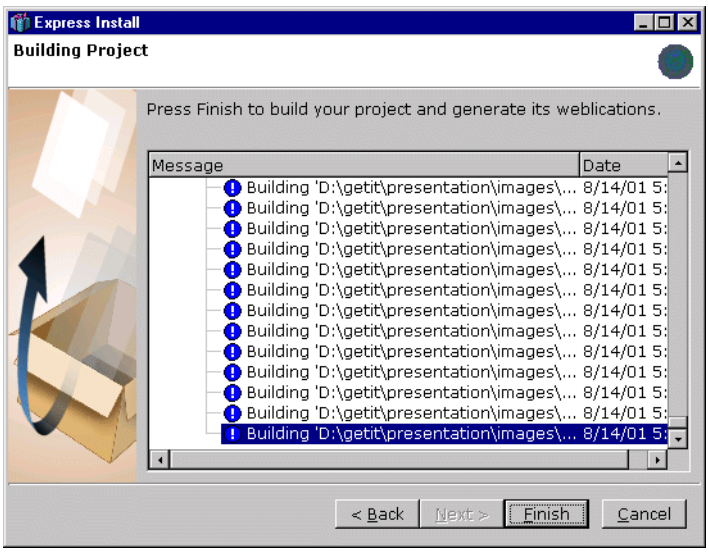

Fig. 2.31 Building the project

When the process is complete, a message is displayed at the bottom of the dialog box that the compilation succeeded.

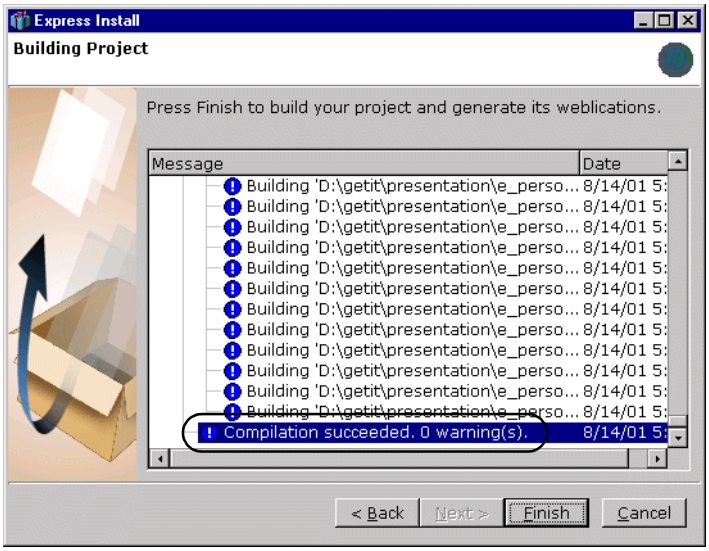

Fig. 2.32 Compilation completed

6. Click **Finish**.

Installation of the weblications is now complete.

Installing Get.It!—Solaris or Linux with JRun

The installation of Get.It! 2.0.1 on Solaris or Linux must first be completed on a Windows system. The WAR file created during this process is then copied to your Solaris or Linux system.

The following procedure outlines the installation procedure for Get.It! on a Solaris platform with JRun as the application server. The installation on Linux is similar, with special instructions for Linux noted in the procedure.

Building the Files on Windows

1. Install Get.It! using the installation procedure beginning on page 2-4. During the installation, do the following:

a. When the message is displayed asking if you want to exit and install JRun first, click **No**.

b. At the end of the installation, deselect the option to log in to the Get.It! Administration, and then click **Finish**.

2. After the installation is complete, open a Command Prompt and change directories to the location of the Get.It! bin directory. The default location is C:\getit\bin.

3. At the prompt, type `deploy solaris` (or `deploy linux`), and then press Enter.

   Prompts are displayed indicating that the presentation and binary files are being copied. A deploy directory is created in the getit directory.

4. Type `deploy war`, and then press Enter.

   This will create a **getit.war** file, which contains the contents of the deploy directory created earlier.

5. Transfer the **getit.war** file to any directory on your application server.

Configuring the Installation on Solaris or Linux

1. Open the JRun Management Console and log in.
2. On the menu at the left, select **JRun Default Server>Java Settings**.
3. Click **Library Path**.
4. Add the path to the AssetCenter bin directory and the Oracle lib directory. On Solaris, also add /usr/bin and /usr/ucblib to the path.
5. Click **update**. The updated path is shown in Figure 2.33.

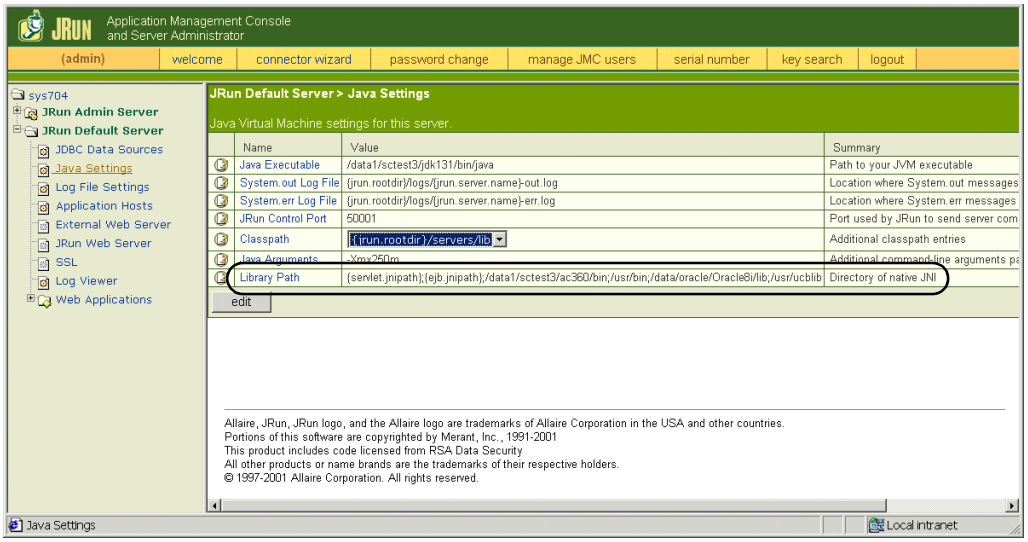

Fig. 2.33 Set the Library Path

6. Click **Web Applications**.

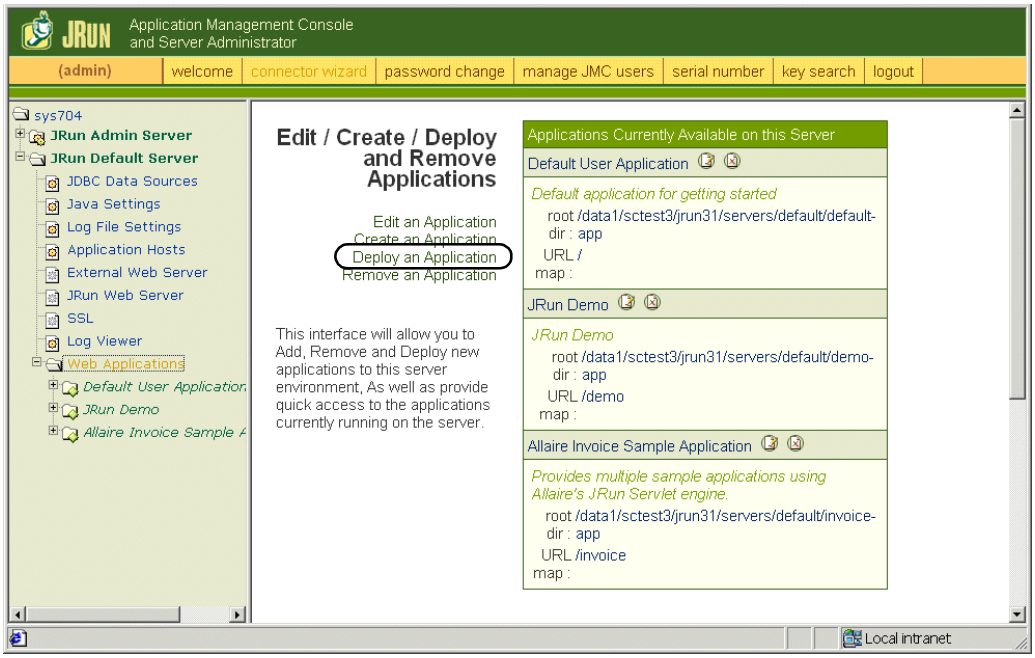

Fig. 2.34 Deploy the application

7. Click **Deploy an Application**.
8. Fill in the form as follows (the completed form is shown in Figure 2.35):
   a. Browse to the location of the WAR file created earlier.
   b. Select JRun Default Server.
   c. Application Name = getit
   d. Application URL = /getit
   e. Make note of the path defined for the Application Deploy Directory. You will need this information later.

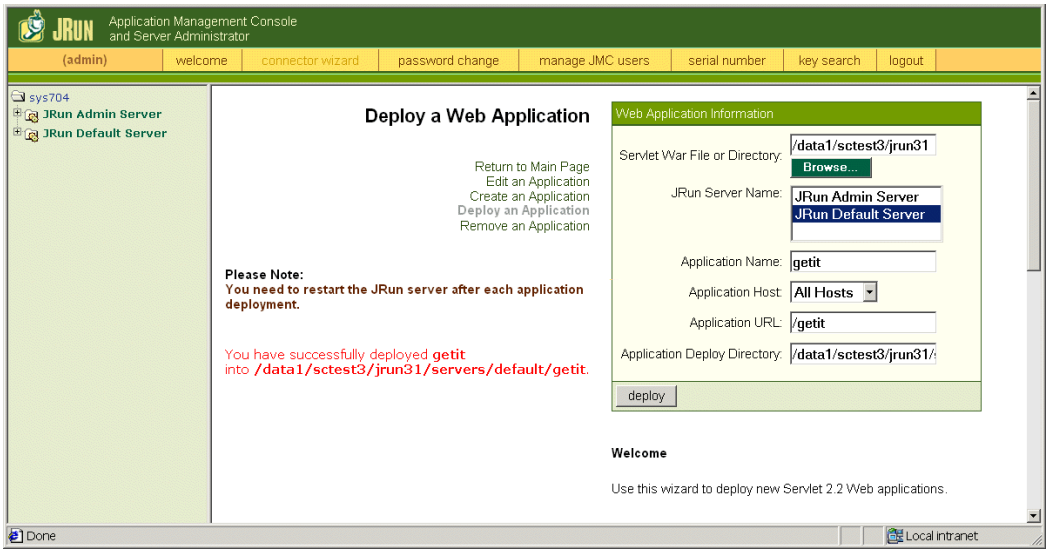

Fig. 2.35 Set the Application Name and URL

9. Click **deploy**. This will unzip the WAR file into the Application Deploy directory.
10. When the screen is displayed indicating that the application was successfully deployed, select **JRun Default Server>Web Applications>GetIt>Servlet Definitions**.

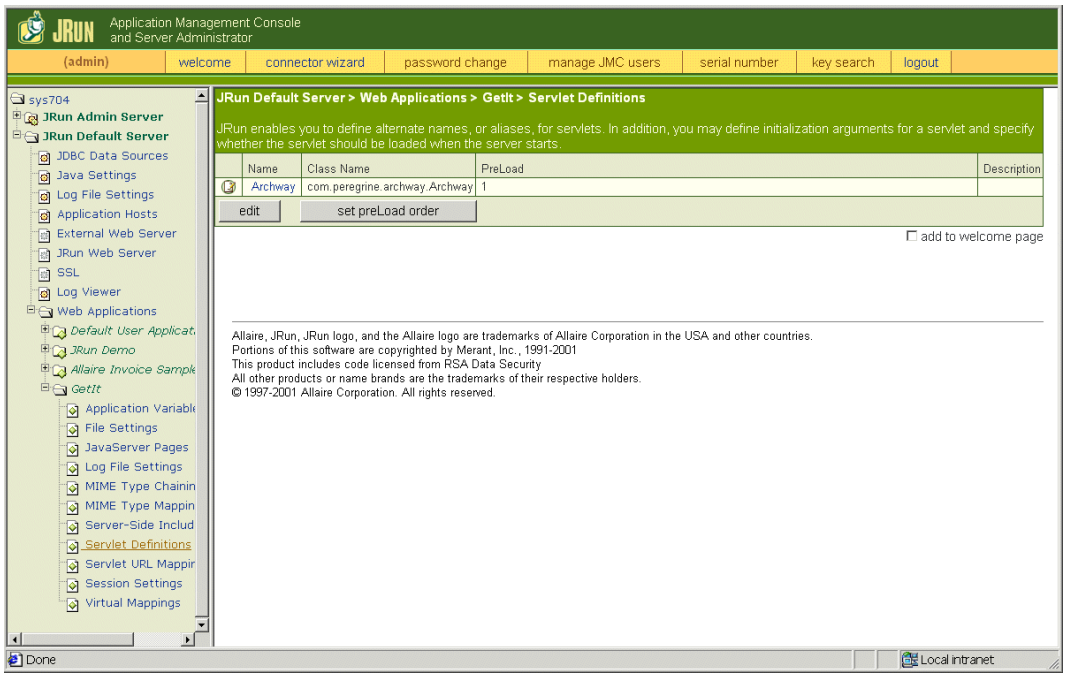

Fig. 2.36 Edit the GETIT_HOME parameter

11. Click **edit**.
12. In the edit window displayed, scroll to the right. Edit the GETIT_HOME Init Argument, replacing YOUR_VALUE_HERE with the directory you noted in step 8e (see Figure 2.37).

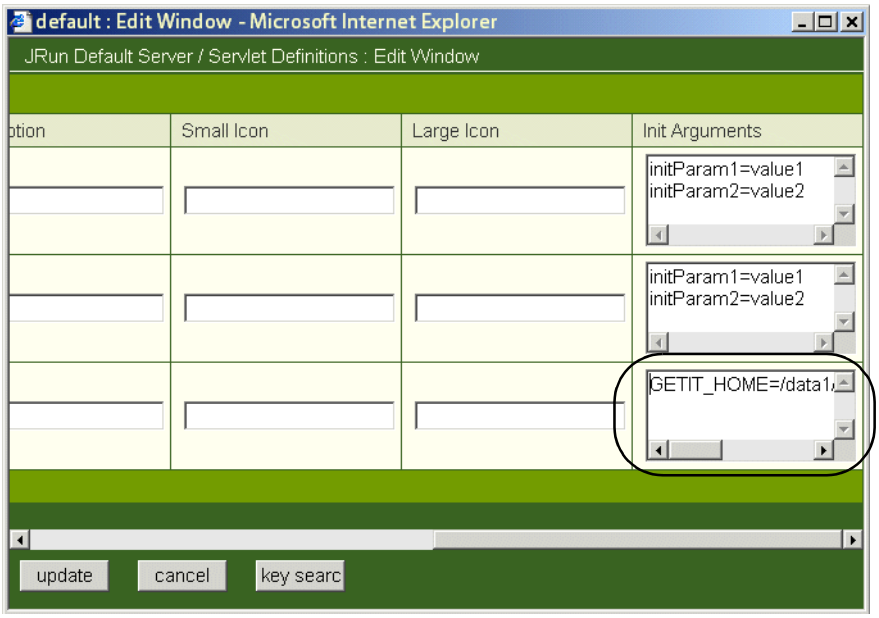

Fig. 2.37 Edit Window

13. Click **update**, and then log out of the Management Console.
14. In a Command Prompt on your Solaris (or Linux) system, change directories to the Application Deploy Directory created in JRun in step 8e.

    Type the following commands, pressing Enter after each command:

    ```
    mv presentation/* .
    rmdir presentation
    ln -s . presentation
    ```
15. To stop and restart the JRun Default Server, type the following commands from the jrun/bin directory:

    ```
    ./jrun -stop default &
    ./jrun -start default &
    ```
16. Make sure that your Web server has a getit virtual directory pointing to the Application Deploy Directory noted in step 8e.
17. To log in to Get.It!, open a Web browser and type:

    ```
    <servername>/getit/login.jsp
    ```

    or, to log in to the Admin module directly, type:

    ```
    <servername>/getit/admin.jsp
    ```

Configuring the AssetCenter amdb.ini File

After installing Get.It! on a Solaris or Linux machine, perform the following steps to establish a connection between the AssetCenter client libraries and the target database:

1. Connect to the database with an AssetCenter client on Windows.
2. The **amdb.ini** will now have connection parameters for that database. Copy the file to the Solaris or Linux system.
3. Move the file to the home directory of the owner of the AssetCenter client libraries.

This will perform establish an encrypted password and generate a valid **amdb.ini** file.

Using the Admin Module to Configure Get.It!

After installing Get.It! on Windows, the Admin weblication is launched to allow you to define appropriate settings for connecting to your servers.

The first page displayed is a login page. Enter an administrator ID and password and click **Login as Administrator**. If you are not ready to set up an Administrator ID and password, click **Login as Administrator** without entering any information.

After you have logged in, error messages will display because you have not yet configured Get.It! for the database you are using. If you are using AssetCenter only, you will need to remove all of the settings that configure Get.It! for the SCAdapter. If you are using ServiceCenter only, you will need to remove the ACAdapter-specific settings.

The first screen you see, the Control Panel, shows the connections between Get.It! and the back-end systems. You can use this screen after you have configured Get.It! to check the status of your database connections and to reset the server.

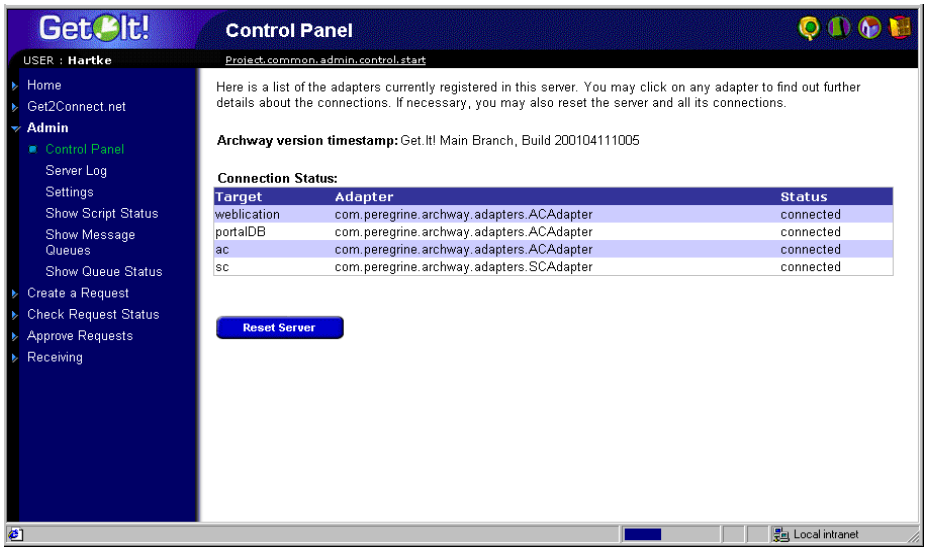

Fig. 2.38 Admin Module—Control Panel

Use the following procedures to configure Get.It! for your AssetCenter and/or ServiceCenter database.

If you are using an AssetCenter database:

1. Log in to the Admin module as an administrator.
2. Click the Settings activity listed in the sidebar.

3. On the Common tab, do the following:

   f. In the AssetCenter Adapter Settings section, Database setting, enter the name of your AssetCenter database (for example, ACDemo351ENG for AssetCenter 3.51).

   g. If you are not using ServiceCenter, make the following change:

      In the General Execution Options section, Adapters setting, delete “sc=SCAdapter”.

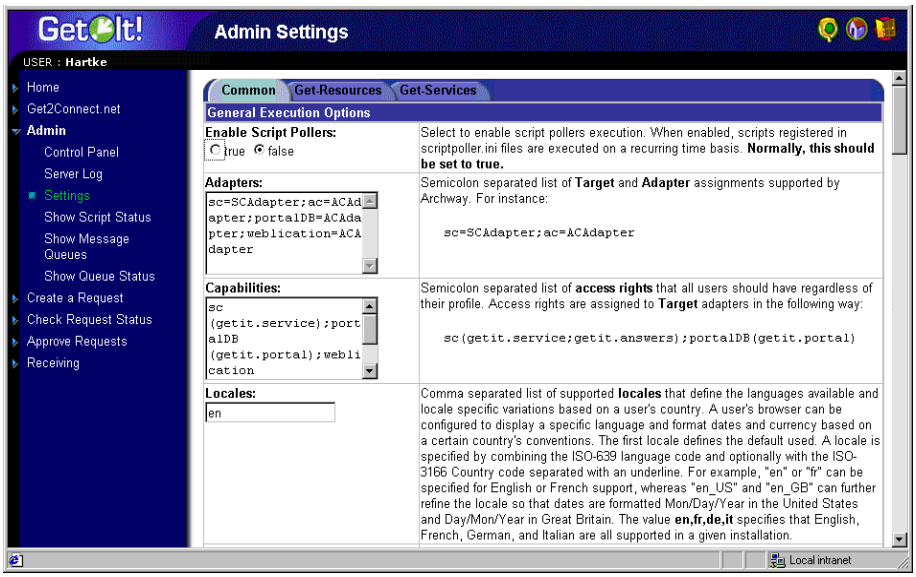

Fig. 2.39 Admin Module—Settings

4. Scroll to the bottom of the page and click **Save**.

   The Admin Control Panel is displayed because the server must be reset to bring the changes into your system.

5. Click **Reset Server**.

If you are using a ServiceCenter database:

1. Log in to the Admin module as an adminstrator.
2. Click the Settings activity listed in the sidebar.
3. If you are not using AssetCenter, make the following changes:

   On the Common tab, in the General Executions Options section, Adapters, do the following:

   a. Delete “ac=ACAdapter”.

   b. Change “portal DB=ACAdapter” to “portal DB=SCAdapter”.

c. Change “weblication=ACAdapter” to “weblication=SCAdapter”.

4. In the ServiceCenter Adapters section, update the Host and Port settings for your ServiceCenter server.
5. Scroll to the bottom of the page and click **Save**.

   The Admin Control Panel is displayed because the server must be reset to bring the changes into your system.
6. Click **Reset Server**.

Continue to the following sections as needed to complete the configuration of your installation.

Loading the ServiceCenter Files

If you will be using ServiceCenter as a back-end system for Get.It!, you will need to load the ServiceCenter formatted files included with the installation of Get.It! onto your ServiceCenter system. These files are used by ServiceCenter event services and contain the mappings for events from Get.It! to ServiceCenter. For example, when a ticket is opened in Get.It!, the fields are mapped from Get.It! into the appropriate fields in ServiceCenter where the data will be stored.

The process includes:

- Loading the files.
- Stopping and restarting the ServiceCenter service or running an initer system command to complete the update to the ServiceCenter files.

The procedure is the same whether you are using ServiceCenter 4.0 or ServiceCenter 3.0x. However, the files you will be loading are different for ServiceCenter 4.0.

The following table shows the files to load depending on the version of ServiceCenter you are using.

ServiceCenter 3.0x	**ServiceCenter 4.0**
...getit\packages\common\common\config **portal.unl** **qman.unl**	...getit\packages\common\common\config **portal.unl** **qman.unl**

ServiceCenter 3.0x	ServiceCenter 4.0
...getit\packages\GetService 2.0\ service\config **callmgmt.unl** **epmx.unl** **smes.unl**	...getit\packages\GetService 2.0\ service\config\Version4 **callmgmt.unl** **epmx.unl** **axcessm.unl**

The following procedure includes the process for loading the **portal.unl** file. Repeat steps 4 through 6 to for *each* file you will need.

To load a file into ServiceCenter:

1. Verify that the ServiceCenter server is running.
2. Open a full (system) client and log into the system as an administrator.
3. From the Toolkit tab, select **Database Manager**.
4. From the **Options** menu, select **Import/Load**.

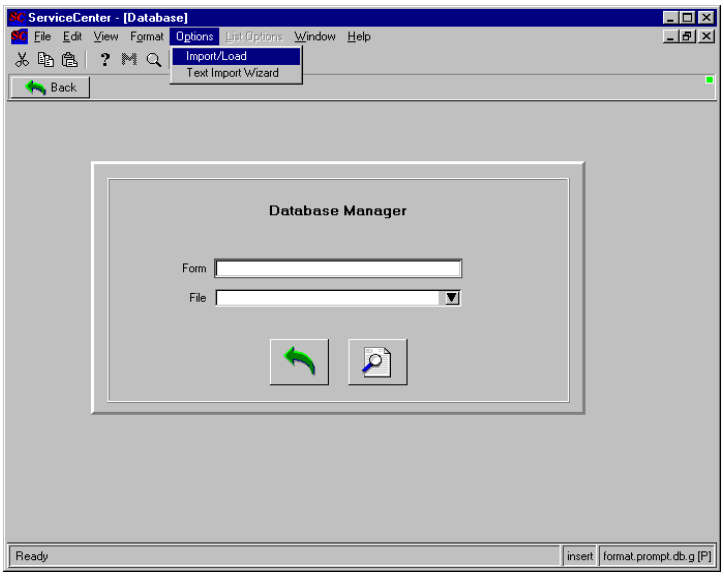

Fig. 2.40 ServiceCenter file import

5. Enter the full file name and path where you installed the Get.It! files. The default location is:

   C:\getit\packages\common\common\config\portal.unl

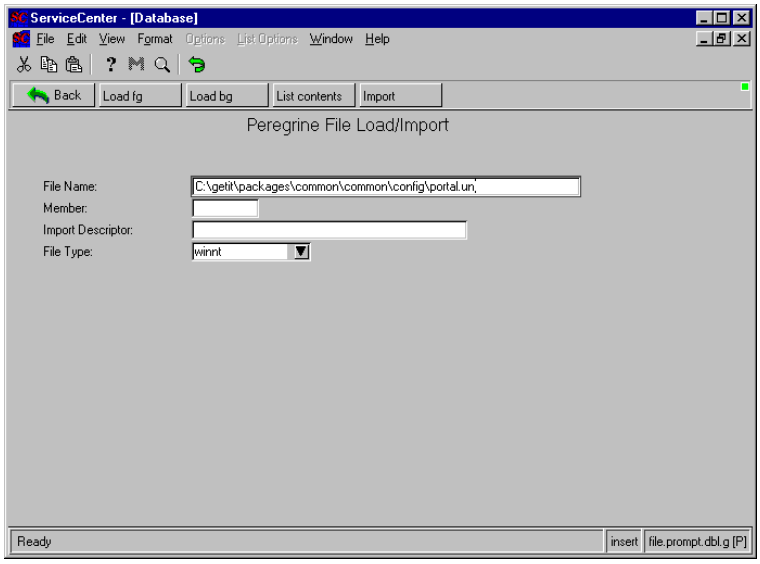

Fig. 2.41 Loading the files into ServiceCenter

6. Click **Load fg**.

   You are returned to the Database Manager form. The following message is displayed at the bottom of the form:

   C:\getit\packages\common\common\config\portal.unl file loaded.

7. Repeat steps 4-6 to load each file.
8. Click **Back** to return to the home menu.
9. Do one of the following:
   - If you are able to stop and start the ServiceCenter service without impacting other users, log out of the client and then cycle the ServiceCenter service.
   - If you are using one of multiple clients pointing to one ServiceCenter server and cannot stop and start the ServiceCenter service, do the following:
     a. Click **Command**, to open a command prompt.
     b. Type `*aapm.global.initer`, and then press ENTER.
     c. Type `*aapm.server.initer`, and then press ENTER.
     d. Click **Back**, and log out of the client.

Configuring Get.It! to use the Demo Weblication

If you have installed the Get.It! demo weblication, do the following to access the weblication:

- Configure a DSN for the ODBC connection.
- Set up a JDBC adapter between Get.It! and the sample MS Access database included with the weblication.
- Use the Admin module to add the JDBC adapter to the list of adapter parameters in the **archway.ini** file.

Instructions for completing these tasks are included in the JDBC Adapter section in Chapter 7, "Adapters".

Server Processing Considerations — Multiple JVMs

Note: Multiple JVMs are supported only if you are using JRun as your application server with IIS as the Web server.

After completing your installation of Get.It!, decide whether or not you will need multiple Java virtual machines (JVMs).

A Get.It! server running a weblication such as Peregrine's Get.Service! or Get.Resources! typically consumes 60 to 100 MB of memory per Archway process. (Get.Resources! skews towards the high end, Get.Service! towards the lower).

Typically, an Archway process of this type is used by all concurrent Web sessions. However, it is possible to add JRun servers with a variable number of processes, using an out of the box "round-robin" load-balancing JSP page that spreads sessions across the processes on a single machine. To take advantage of this you will need a multi-processor Web server and you will need to configure multiple JVMs and multiple virtual Web directories.

Each Archway process creates a single AssetCenter database connection and/or a single ServiceCenter client/server connection. Therefore, the memory consumed on the AssetCenter database server is the same as that consumed by a single client connection. The memory consumed on the ServiceCenter server is the same as that of a single ServiceCenter server process. The need for extra Archway instances (JVMs) is directly related to three variables:

- The number of concurrent users.
- The processing power of the machine hosting the Get.It! Web server.
- The number of processors on the machine.

Optimal execution is achieved when the number of processes is increased to match the number of processors on a machine. Low-end processors, for example, a Pentium 450, should be capable of producing acceptable load handling for around 100 concurrent sessions on a single Archway process.

Note that memory usage does not increase significantly per session, because the architecture is based on the sharing of a set of resources and database connections among all sessions handled by the same Archway process.

However, the small amount of session-specific memory is released as the users log off or as their sessions expire. Typically, expiration occurs after five or ten minutes of inactivity, but this is configurable. If you configure multiple JVMs, you may need to change the javaargs parameter to define the size of heap memory given to the JVMs. For more information, refer to "Setting Heap Memory for Single or Multiple JVMs" on page 2-48.

Before you begin the procedure for configuring multiple JVMs, make sure that you have completed the installation of Get.It! 2.0.1 and JRun 3.1 using the procedures at the beginning of this chapter. There are four parts to the configuration of multiple JVMs:

- Adding the extra JRun servers.
- Adding the additional JVM connectors.
- Adding and configuring the extra virtual directories to your Web server.
- Updating the **archway.ini** file with new parameters.

Adding a JRun Server

The following procedure includes instructions for creating one JRun server. Repeat the procedure to add as many servers as you will need.

To add a JRun server:

1. Stop the JRun Default server.
2. Copy the /servers/default directory to a directory with the new server name vm1.
3. Open the **local.properties** file in the new directory (/servers/vm1). The **local.properties** file defines the server-specific settings of a Jrun server. It includes basic properties such as name and port settings, as well as the applications that are deployed on the server.
4. Make the following changes to the new **local.properties** file:
   a. Change the jrun.server.displayname property to reflect the new name (“ VM1”) of the server. This name will appear in the JMC. The code should now look something like this:

```
## jvm properties
jrun.server.displayname=VM1 Server
```

b. Remove applications other than the default-app in the Web Application Settings section. If you have not added any applications to the default server, you do not need to delete anything other than the demo-app.

   For example, each application might have a section that looks like this:

   ```
   jmc-app.rootdir=C:\\JRun\\servers _\\admin\\jmc-app
   jmc-app.class={webapp.service-class}
   webapp.mapping./=jmc-app
   ```

   Delete that application's settings unless it refers to the default-app.

c. Add the following line in the Web Services section below the existing docroot definition.

   In the example below, <getit_root> should be replaced with the absolute path to the Get.It! directory that each JVM will be pointing to:

   ```
   web.paths./getit=<getit_root>/presentation
   ```

d. In the Web Application Settings section, reset the servlet.webapps property to contain only the default application. The line should now look like this:

   ```
   servlet.webapps=default-app
   ```

e. Disable session persistence.

   In the Session Services section:

   ```
   session.persistence=false
   ```

f. Change the port settings to unique ports.

   In the Web Services section:

   ```
   web.endpoint.main.port=8101
   ```

   In the Control Services section:

   ```
   control.endpoint.main.port=53001
   ```

   In the jcpservices section:

   ```
   jcp.endpoint.main.port=8084
   ```

   In the EJB Properties section, under "extra virtual machine args:"

   ```
   -Dejipt.classServer.port=2324
   -Dejipt.homePort=2334
   ```

Note: Although these last two properties are not used by Get.It!, they must be changed to avoid socket binding issues.

g. Save the **local.properties** file.

5. If you have added any applications to the default server, delete the directories of all applications other than the default-app from /servers/vm1.
6. Open the **jvms.properties** file in the <JRun_rootdir>/lib directory. Jrun uses the **jvms.properties** file to determine what servers should be instantiated. Add a line for the new server. On a Windows NT system, the file should appear as follows:

   ```
   admin=C:/JRun/servers/admin
   default=C:/JRun/servers/default
   vm1=C:/JRun/servers/vm1
   ```

   Note that the server name appears at the beginning of the line, and the directory name appears at the end of the line.
7. To start the new JRun server:
   a. Add the new server as a Windows NT service using the command-line utility:

      ```
      drive:\<jrunroot>\jrun -install "VM1 Service" vm1 -quiet
      ```

   b. Go to the Services panel (**Start>Settings>Control Panel**) and find the new server name (VM1 Server in this example). Click **Start**.

   You can also start the new server as an application:

   a. In the <JRun_rootdir>/bin directory, copy the **jrun-default.bat** file to **jrun-vm1.bat**.
   b. Edit **jrun-vm1.bat** to specify the name of the new server:

      ```
      @echo off
      start jrun -start vm1
      ```

   c. Run the **jrun-vm1.bat** file.
8. To add another server, follow steps 1 through 7, naming the next server vm2. In step 3.f, increment the port numbers by one. For vm2, the new settings would look like this:

   ```
   web.endpoint.main.port=8102
   control.endpoint.main.port=53002
   jcp.endpoint.main.port=8085
   -Dejipt.classServer.port=2325
   ```

```
-Dejipt.homePort=2335
```

Adding Multiple JVM Connectors

For each JVM you create, run the JRun Connector Wizard so that the virtual directories you will be creating in your Web server can connect to the individual JVMs. For this to work, you must first **remove the global JRun connector filter** installed during the initial installation of Get.It! and JRun.

To remove the global connector filter:

1. Start the Microsoft Internet Services Manager.
2. Right-click on the icon with the computer's name, and select Properties.
3. Edit the Master Properties in the IIS tab.
4. Select the ISAPI Filters tab.
5. Select the entry for JRun and click **Remove**.
6. Click **OK**, and then close ISM.
7. Stop all IIS services (**Start>Settings>Control Panel>Services**).
8. Find the Inetpub/scripts directory and remove the following files: **jrun.dll**, **jrun.ini**, and any jrun log files. Save the log files to another directory if you need to keep the old data.

Next you need to install a connector for each JVM you created.

To install a connector:

1. In your Inetpub/Scripts directory, create a directory for each JVM you created in JRun. Also create one for your master Get.It! JVM (JRun Default Server). You should name the directories in accordance with the JVMs. Example: getit, getit_vm1, getit_vm2, getit_vm3, etc.
2. Start the JRun Management Console (**Start>Programs>JRun 3.1**) and log in.
3. Run the Connector Wizard as you did when JRun was first installed, but create a connector for each JVM. Do not create one for the JRun Admin Server. Instructions for using the JRun Connector Wizard begin on page 2-13.
    a. Select a JVM and follow the instructions for step 1 of the Connector Wizard.
    b. In step 2, make sure each JVM has a unique Port number for the "JRun Server Connector Port."

c. In step 3, select the appropriate directory you created in the Inetpub/scripts directory. **It is very important that you de-select the "Install as a Global Filter" option**. If this is selected, the Global filter and the individual filters will cause an error and the IIS server will not connect to JRun.

d. In step 4, proceed to the finish.

Important: Interference will occur if **jrun.dll** files are kept in the root scripts directory. Check the scripts directory and remove any **jrun.dll** files that are there.

Updating the JVM Script Path

After you have run the Connector Wizard for each JVM, you will need to update the scriptpath parameter in the **jrun.ini** file for each connector you created to reflect the actual location of the JRun DLL in the ...Inetpub\scripts subdirectory. The Connector will not add the subdirectory you created in step 1 above.

For example, if you created a subdirectory called "getitmain", the scriptpath would be:

```
scriptpath=/scripts/getitmain/jrun.dll
```

Configuring Multiple Virtual Directories

The next step is to create the Web server's virtual directories.

- Go to the IIS Microsoft Internet Services Manager and create each virtual directory. Make sure the prefix of each directory name is identical and is the same name used in creating the JVM's "web.paths" property (see step 4c in "Adding a JRun Server" on page 2-42).

  Example virtual directories: getit_vm1, getit_vm2, getit_vm3, and so on.

  where "getit" is the prefix name, "vm" is the suffix name, and "1" is the increment. These are explained in the "Updating the archway.ini" section on page 2-47 and can be used to customize the installation.

After you have created the multiple virtual directories, use the instructions in this section to configure them.

1. Start the Microsoft Internet Services Manager from the Start menu.
2. Right-click the first virtual directory you just created and select **Properties**.
3. In the Application Settings area of the window you should see the virtual directory you selected in the *Name* field. Click **Configuration**. If there is no application configured, click the **Create** button.

4. On the APP Mappings tab, click **Add**.
5. For the Executable, add the path or browse to and select the appropriate connector filter you created in the Inetpub/scripts directory. Add **.jsp** for the extension.

   Example: For virtual directory “getit_vm1” the path would be:

   ```
   C:\Inetpub\scripts\getit_vm1\jrun.dll
   ```
6. Repeat this process for each virtual directory you created.

The server does not need to be rebooted.

Updating the archway.ini File

In order to complete the configuration of multiple JVMs, the following parameter must exist in the **archway.ini** file. This file is located in the ...\getit\bin directory.

- **jrunroot—** This parameter points to the Java server’s directory under the JRun installation. It is a minimum requirement for initializing multiple JVMs. The parameter is set during the JRun installation. Verify that the path is C:\JRun\servers. The default if not set is c:/Program Files/Allaire/JRun/servers. Example: `jrunroot=C:\Jrun\servers`.

The following settings are optional. They allow you to customize the JVM’s naming convention, the type of JVM to use, the JVM port, and a JVM on/off switch. This would be desirable for configuring multiple versions of Get.It! to run at the same time on the same server. These settings are also useful for creating a Production and Test environment on a single server.

These settings are for advanced users and systems. You should be very familiar with the multiple JVM environment before you attempt to customize at this level.

- **jrunport** — If you choose to use your Java server’s internal WWW server and multiple JVMs, this parameter specifies the TCP port on which the JRun Web server listens for HTTP requests from the browser. The default value is 8100. This port number is incremented by one for each additional server you have configured. If you choose to have the default port different from 8100, you will need to specify `jrunport=<new_jrun_port>`. Get.It! will calculate the ports for the additional servers by incrementing the default value by one for each additional server.
- **PrefixURL —** Sets the Web or Java server URL (http://<webserver>/"URL"/login.jsp). This needs to be set if the default Get.It! virtual directories are named something other than "getit". This would be

useful in creating multiple JVMs using multiple versions of Get.It!. The Java server and the WWW server must have matching virtual directory names (refer to “Adding a JRun Server” and “Configuring Multiple Virtual Directories”). The default setting is `PrefixURL=getit`.

- **SuffixURL** — Name for the Java Server JVM directories and Get.It! Web server URL suffix (http://<webserver>/getit_"URL"1/login.jsp). The default setting is `SuffixURL=vm`.
- **WebSvrVMStart** — Sets the increment start number for the WWW server vm (.../getit_vm(i)/login.jsp). The default setting is `WebSvrVMStart=0`.
- **VMTypeJrun** — Setting this to True tells the load balancer to use the internal WWW server of your Java Server (for example, JRun or Tomcat). If you choose this option, there are several limiting issues to be considered. The default setting is `VMTypeJrun=false`.
- **UseJVM** — Enables you to turn off/on Round Robin load balancing after all JVMs have been configured. Since the load balancing process is automatic after the configuration, you may need at some point to turn this off for debug purposes. This lets you keep all the JVM configurations without modification or deletion. The default setting is `VMTypeJrun=true`.

After you have added the JRun servers, the Web server virtual directories, and updated the **archway.ini** file, restart the JRun services and start the IIS Web services. It is not necessary to restart the system.

Single server, multi-JVM load balancing is automatically used when accessing Get.It! using the **index.jsp** provided. The default URL is: `http://<webserver>/getit/index.jsp`. The request will be redirected to the Web server or the JRun servers depending on your settings. The **index.jsp** file can be used to access Get.It! even if load balancing is not configured. This will automatically forward you to the login screen (**login.jsp**).

Setting Heap Memory for Single or Multiple JVMs

Depending on the size of your system, you may need to define the size of heap memory given to the JVMs. The parameter, javaargs, is used to accomplish this and is set using the JRun Management Console. This may need to be done for single as well as for multiple JVMs. If you have configured multiple JVMs, you will need to adjust the heap memory only for the additional JVMs you have added. You do not need to adjust memory for the default.

The following example setting provides 256 MB. Larger settings may be appropriate on larger systems.

```
user.javaargs=-Xmx256M
```

To set the javaargs parameter:

1. Open the JRun Management Console (**Start>Programs>JRun 3.1>JRun Management Console**).

   This launches the Management Console in a Web browser and prompts you for your user name and password, established during the JRun installation.

2. Type your user name and password, and then click **login**.

3. In the screen displayed, open the JRun Default Server directory, and then click **Java Settings**.

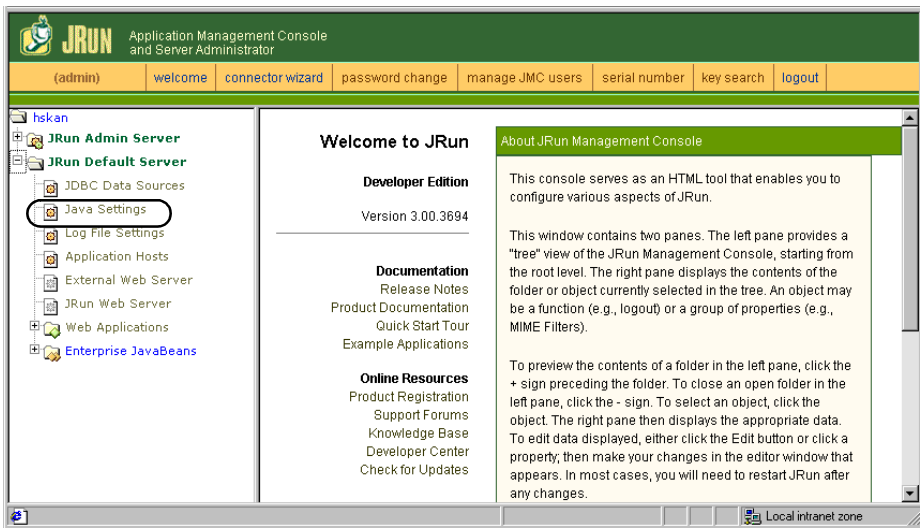

Fig. 2.42 JRun Management Console

4. In the list of Java Virtual Machine settings, click **Java Arguments**.

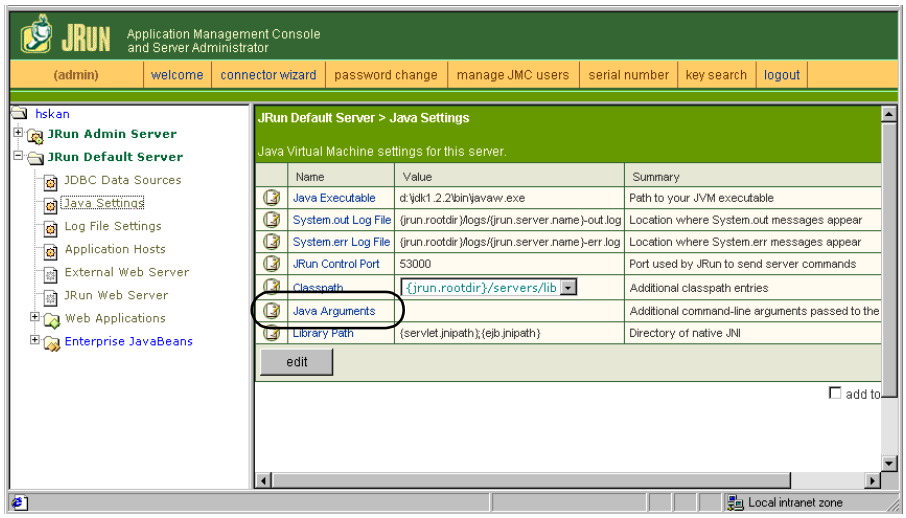

Fig. 2.43 JRun Management Console—Java Arguments

5. In the edit dialog box displayed, type the parameter you want to set, using the following format. This parameter is case sensitive.

   ```
   -Xmx256M
   ```

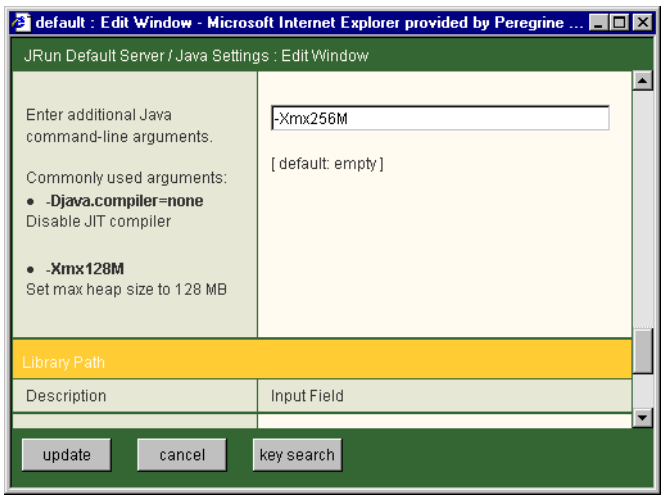

Fig. 2.44 JRun Management Console—Edit Window

6. Click **update**.

   You are returned to the main Management Console screen.

7. Log out of the JRun Management Console.

Setting Parameters for the Production Environment

Consideration should be given to establishing initialization and JRun settings to ensure that your installation runs smoothly in a production environment. The following table describes these settings and gives the default or recommended value for each parameter.

All of the parameters are set in the **archway.ini** file, except javaargs. Refer to "Setting Heap Memory for Single or Multiple JVMs" on page 2-48 for instructions for setting this parameter.

Parameter	Description	Value
user.javaargs	Defines the size of heap memory given to JRun's Java Virtual Machines. Refer to "Server Processing Considerations — Multiple JVMs" on page 2-41 for more information.	Varies according to the size of your system. A typical value would be: userjavaargs=-Xmx128M which provides 128 MB. Larger settings may be necessary on larger systems.
scripttimeout	Specifies the number of seconds given to each user to execute a script. If a user script runs longer than the maximum time-out value, the script operations are aborted. Proper setting of this option is important to prevent runaway scripts from tying up all user sessions.	The default setting is: scripttimeout=30
logstdout	When set to false, prevents duplicate logging sent to stdout.	logstdout=false
sessiontimeout	Specifies the number of seconds to allow inactive sessions to exist before they are expired. This setting is important because it ensures proper cleanup of old sessions that may potentially be holding on to needed memory resources.	The default setting (in seconds) is: sessiontimeout=600

Parameter	Description	Value
maxscriptrunners	Controls the size of the FESI ScriptRunner pool created for Archway.	This parameter should rarely need to be set greater than 10 (the default): maxscriptrunners=10 Each ScriptRunner can consume a significant amount of memory, yet the efficiency of the system does not increase with a number of script runners greater than 10.
maxquerycount	If this parameter is set, it limits the number of records that Archway will fetch for any given query. It is a good idea to set this parameter to prevent runaway queries or bad scripts from bringing thousands of records into a single XML response. In some cases, a system does need to process a large number of records. The recommendation in such cases is to perform several queries for smaller record chunks. The Start and Count parameters to sendDocQuery() help accomplish this.	Depends on the limits you want to set for your system.

Internationalization

This section includes information for Get.It! installations using languages other than English. It also includes tables of supported country and currency codes.

Setting the Content Type Encoding

A variable can be set in Get.It! Studio to specify the character encoding used in the JSPs (Compilation>Project Settings>Build Variables). Two procedures are necessary:

- Define the settings on the Admin Settings page.
- Define the settings and build using Get.It! Studio.

The following settings are available:

- United States and Europe—ISO-8859-1, the default value for Windows.

- Japanese (Windows only)—Shift_JIS
- Polish (Windows only)—ISO-8859-2

To set the encoding:

1. Log in to Get.It! as an administrator and click the Admin module. Select the **Settings** page from the activity menu.
2. On the Common tab, in the General Execution Options section, change the Content type encoding field to the desired setting.
3. Click **Save** at the bottom of the form.
4. Reset the server from the Admin Console form.

Important:	These settings must also be defined and built in Get.It! Studio by modifying the build variables (Compilation>Project Settings>Build Variables). Refer to the *Get.It! Tailoring Guide* for instructions.

Setting the Locale Code

When a user logs on to Get.It!, the logon programs detect the preferred language set in the browser and determine if the user's language is supported by Get.It!. If the language is supported, Get.It! will be displayed in that language.

If the language is not supported, the default language is used. The default language is set in the **archway.ini** file through the Locales field in the Admin Module. The first entry in this field is the default language.

The following languages with their country codes are supported in Get.It! 2.0.1:

Language	Country Code
English	en
French	fr
German	de
Italian	it
Japanese	ja

Language	Country Code
Polish	pl
Spanish	es

The following procedure includes instructions for setting the Locale code. You can use the same procedure for setting the Currency code.

To change the language you are using with Get.It!:

1. Open the Admin Settings page.
2. In the Locales field, type the two-letter designation of the language you want to use.

Note: If you would like a drop-down list from which users can choose the language they want to use, include English (en) in your list of languages in the Locales field. Include all the languages in this field that you want to have supported, separated by commas. Figure 2.45 shows settings for English, French, German, and Italian locales.

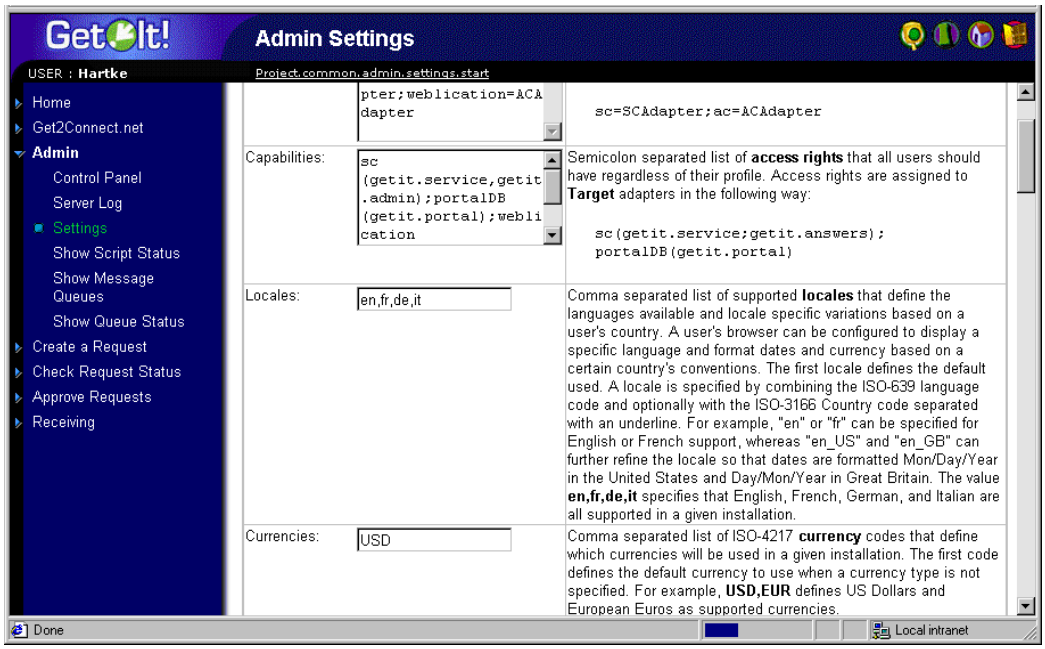

Fig. 2.45 Admin Settings—Locales and Currency

3. Scroll to the bottom of the Settings page and click **Save**.

The Control Panel is displayed.

4. Click **Reset Server** to reset the connection to the back-end systems and import the new information.

   If you have set just one language in the Locales field, all window content will be displayed in that language.

   If you have set multiple languages, the login screen will be displayed in English with a language field and a drop-down list from which the user can select a language, as shown in Figure 2.46. However, if the back-end system is set up to use a different language than Get.It!, data from that system will be shown in the language supplied by that system.

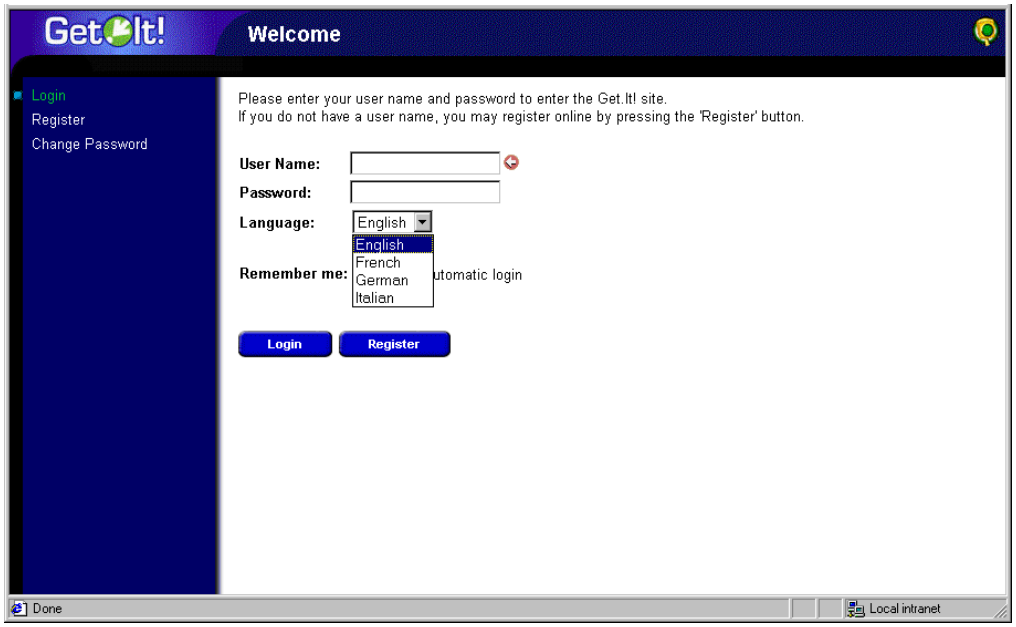

Fig. 2.46 Language drop-down list

Setting the Currency Code

The following chart shows the ISO codes for the currencies supported in Get.It! 2.0.1. These codes are entered in the “Currencies” fields in the Admin Settings page of the Admin module.

Currency Code	Currency Name
AED	United Arab Emirates Dirham
ALL	Albania, Lek
ARP	Argentine Peso
ATS	Austria, Schilling
AUD	Australian Dollar
BEF	Belgian Franc
BGL	Bulgaria, Lev
BHD	Bahraini Dinar
BOB	Bolivia, Boliviano
BRR	Brazilian Real
BYR	Belarussian Ruble
CAD	Canadian Dollar
CHF	Swiss Franc
CLP	Chilean Peso
CNY	China, Yuan Renminbi
COP	Columbian Peso
CRC	Costa Rican Colon
CSK	Czech Koruna
DEM	Germany, Deutsche Mark
DKK	Danish Krone
DOP	Dominican Peso
DZD	Algerian Dinar
ECS	Ecuador, Sucre

Currency Code	Currency Name
EEK	Estonia, Kroon
EGP	Egyptian Pound
ESP	Spanish Peseta
EUR	Euro (Austria, Belgium, Germany, Spain, Finland, France, Ireland, Italy, Luxembourg, Netherlands, Portugal)
FIM	Finland, Markka
FRF	French Franc
GBP	United Kingdom, Pound Sterling
GRD	Greece, Drachma
GTQ	Guatemala, Quetzal
HKD	Hong Kong Dollar
HNL	Honduras, Lempira
HRK	Croatian Kuna
HUF	Hungary, Forint
IEP	Ireland, Punt
ILS	New Israeli Shekel
ISK	Iceland Krona
ITL	Italian Lira
JOD	Jordanian Dinar
JPY	Japan, Yen
KRW	Republic of Korea, Won
KWD	Kuwaiti Dinar
LBP	Lebanese Pound
LTL	Lithuanian Lita
LUF	Luxembourg Franc
LVL	Latvian Lat

Currency Code	Currency Name
LYD	Libyan Dinar
MAD	Moroccan Dirham
MKD	Macedonia, Denar
MXP	Mexican Peso
NIO	Nicaragua, Cordoba Oro
NLG	Netherlands Guilder
NOK	Norwegian Krone
NZD	New Zealand Dollar
OMR	Oman, Sul Rial
PAB	Panama, Balboa
PEN	Peru, Nuevo Sol
PLZ	Poland, Zloty
PTE	Portuguese Escudo
PYG	Paraguay, Guarani
QAR	Qatari Rial
ROL	Romania, Leu
RUR	Russian Ruble
SAR	Saudi Riyal
SDD	Sudanese Dinar
SEK	Swedish Krona
SIT	Slovenia, Tolar
SKK	Slovak Koruna
SVC	El Salvador Colon
SYP	Syrian Pound
THB	Thailand, Baht
TND	Tunisian Dinar
TRL	Turkish Lira

Currency Code	Currency Name
TWD	Taiwan Dollar
UAH	Ukraine, Hryvnia
USD	US Dollar
UYU	Uruguay, Peso Uruguayo
VEB	Venezuela, Bolivar
YER	Yemeni Rial
YUM	Yugoslavia, New Dinar
ZAR	South Africa, Rand

All dates and currency are displayed and edited in the user's language. The formatting is done when the weblication <field> or <input> tags have a type attribute equal to "date" or "currency." Calendars display the translated weekday name and start the week on the user locale's first day of the week.

Currency is displayed using the correct symbol and decimal indicator. However, the decimal point and the thousands separator are determined by the user's preferred language and not the currency being displayed. For example, a French-speaking user will see US dollars as $1 234,00, a German-speaking user will see $1.234,00, and an English-speaking user will see $1,234.00.

Chapter 3
Configuring the Application Servers

This chapter includes instructions for configuring the alternate application servers supported with Get.It! 2.0.1. The following servers are supported:

- WebSphere 4.0
- WebLogic 6.0 SP2
- Tomcat 3.2x

Initial Application Server Configuration

The following procedure is the first part of the configuration process and can be used for WebSphere, WebLogic, or Tomcat. When you have completed this part of the configuration, continue to the appropriate section for specific instructions for the application server you are using.

1. Install Get.It! using the installation procedure in Chapter 2.
2. During the installation, do the following:
   a. When the message is displayed asking if you want to exit and install JRun first, click **No**.
   b. At the end of the installation, deselect the option to log in to the Get.It! Administration, and then click **Finish**.
3. After the installation is complete, open a command prompt window and change directories to the location of the Get.It! bin directory. The default location is C:\getit\bin.
4. At the prompt, type `deploy windows`, and then press Enter.

   If your application server is running on a Solaris or Linux system, type `deploy solaris` or `deploy linux`.

   Prompts are displayed indicating that the presentation and binary files are being copied. A deploy directory is created in the getit directory.
5. In the deploy\WEB-INF directory, open the **web.xml** file using any text editor.
6. Locate the following lines of text:

```
<param-name>GETIT_HOME</param-name>
<param-value>YOUR_VALUE_HERE</param-value>
```

Replace the `YOUR_VALUE_HERE` text with the path to your getit directory on the application server. The default paths are:

Tomcat:

```
C:\Program Files\Apache Group\Tomcat\webapps\getit\
```

WebSphere:

```
C:\WebSphere\AppServer\installedApps\getit.ear\getit.war\
```

WebLogic:

```
C:\bea\wlserver6.0\config\mydomain\applications\getit\
```

7. Save the file.

To complete the configuration of your application server, choose the appropriate instructions for your server from the following procedures.

Tomcat

1. In the command prompt window (in the getit\bin directory), type `deploy war`, and then press Enter.

   This will create a **getit.war** file, which contains the contents of the deploy directory created earlier.

2. Copy the **getit.war** file to the webapps directory at ...\Program Files\Apache Group\Tomcat\.

3. Because Tomcat does not support encryption, copy the following JAR files from the getit/WEB-INF/lib directory to either the Tomcat/lib or JDK/jre/lib/ext directory:

   jce1_2_1.jar
   jcert.jar
   jnet.jar
   jsse.jar
   local_policy.jar
   sunjce_provider.jar
   US_export_policy.jar

   This will enable password encryption in the **archway.ini** file and allow HTTPS encryption when connecting to the B2B server.

4. Restart Tomcat so that the **mod_jk.conf-auto** file will be recreated with the getit information.

5. In the <TOMCAT_HOME>\conf directory, copy **mod_jk.conf-auto** and name the file **mod_jk.conf-getit**.

6. In the **mod_jk.conf-getit** file, change the alias from /getit to /getit/presentation as shown in bold text in the example below. Each command should be on one line.

```
############################################################
# Auto configuration for the /getit context starts.
############################################################
#
# The following line makes apache aware of the location of the /getit
context
#
Alias /getit "C:/Program Files/Apache Group/tomcat/webapps/getit/
presentation"
<Directory "C:/Program Files/Apache Group/tomcat/webapps/getit/
presentation">
     Options Indexes FollowSymLinks
</Directory>
```

7. In the Program Files\Apache Group\Apache\conf directory, open the **httpd.conf** file. Add the following line:

   ```
   Include <TOMCAT_HOME_PATH>/conf/mod_jk.conf-getit
   ```

   If there is an existing line including the **mod_jk.conf-auto** file name, comment it out with a hash (#).

8. Ensure that heap size in Tomcat is set to at least 256 MB. To set this in the global environment:
   a. Right-click on My Computer, and then select **Properties**.
   b. On the Advanced tab, select **Environment Variables**, and then click **New**.
   c. Add the following: TOMCAT_OPTS=-Xmx256M.
9. Restart Tomcat and Apache.

If you need to reinstall Get.It! on Tomcat, first uninstall your current installation by deleting the getit directory from Tomcat. Recopy the **getit.war** file and then restart Tomcat.

WebLogic

1. In the WebLogic applications directory, create a directory called getit.
2. Copy the contents of the deploy directory to the WebLogic applications\getit directory.
3. Go to the Get.It! lib folder (typically wlserver60\config\<my domain>\applications\getit\WEB-INF\lib), where <my domain> is the domain name of the system on which WebLogic is installed.

4. Move the following files to your JRE (or jdk) extension folder (jre\lib\ext):

   sunjce_provider.jar
   jce1_2_1.jar
   local_policy.jar
   US_export_policy.jar

5. Create a virtual directory named getit on your Web server and point it to the WebLogic applications getit\presentation directory.

   This will create a connection between the application and Web servers.

6. Restart WebLogic and the Web server.

If you need to reinstall Get.It! on WebLogic, first uninstall your current installation by deleting the getit directory from WebLogic. Recopy the Get.It! files, and then restart WebLogic and your Web server.

WebSphere

1. In the Command Prompt (in the getit\bin directory), type `deploy war`, and then press Enter.

   This will create a **getit.war** file, which contains the contents of the deploy directory created earlier.

2. Open the WebSphere administrator's console (**Start>Programs>IBM WebSphere>Application Server V4.0**).
3. At the Login screen, click **Submit**. A user name is not necessary.
4. On the menu at the left side of the console, click **Nodes**, **<SystemName>**, and **Enterprise Applications**.

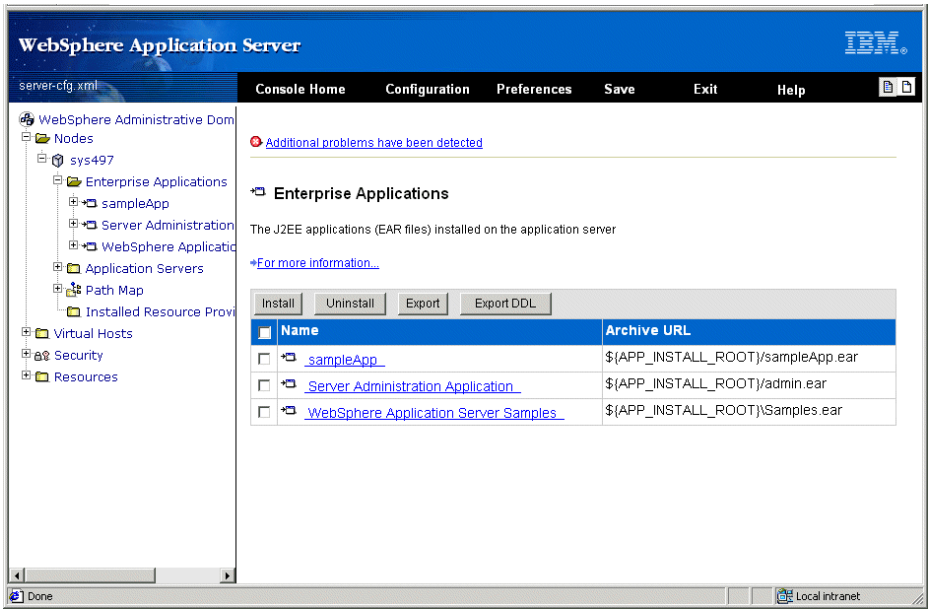

Fig. 3.1 Administrator's console

5. Click **Install**.
6. On the screen displayed, do the following:
   a. Browse to the path to the **getit.war** file. The default is C:\getit\getit.war.
   b. In the Application Name field, type `getit`.
   c. In the Context Root field, type `/getit`.

Note: If you are installing WebSphere on a different machine from your Get.It! installation, fill in the fields in the second half of the screen instead of the top three fields.

The completed form is shown in Figure 3.2.

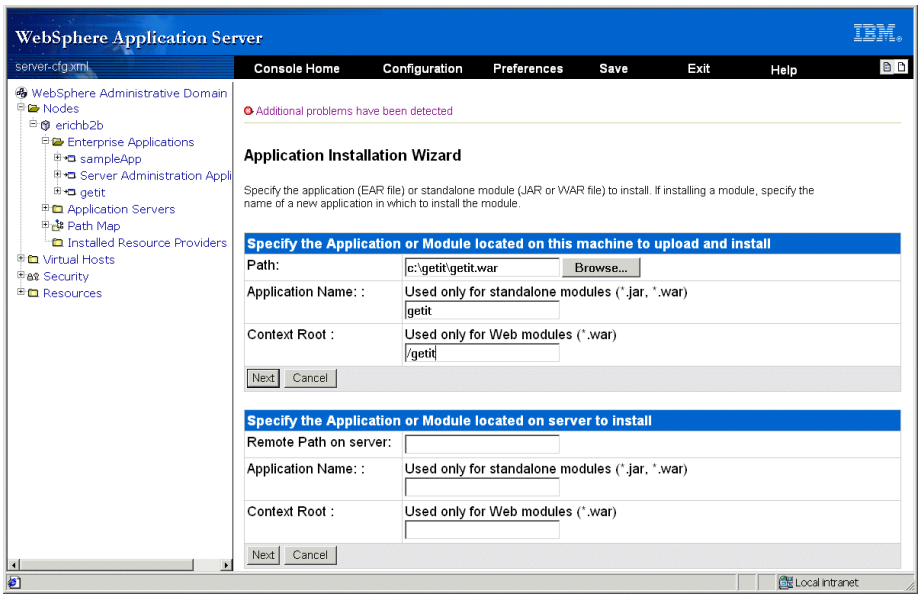

Fig. 3.2 Application Installation Wizard—set Path, Application Name, and Context Root

7. Click **Next**.
8. In the screen displayed, you have the option of selecting to precompile the JSP pages.

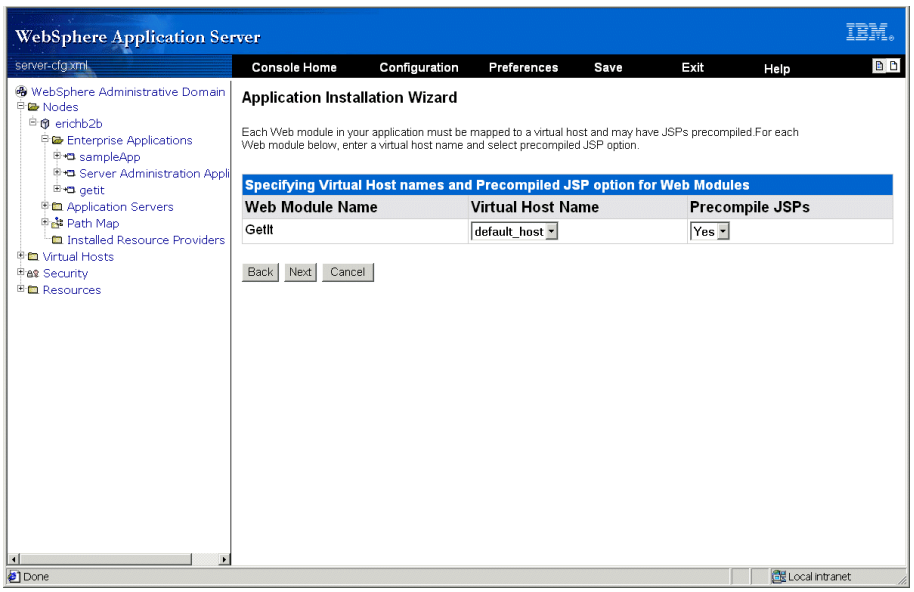

Fig. 3.3 Application Installation Wizard—select to precompile JSPs

For a development system or for testing purposes, it is not necessary to precompile JSPs. For a production system, you should set this option to *Yes* to precompile the JSPs. This process can take several minutes to several hours depending on the memory available on your system and the processing speed. However, if you do not precompile the JSPs, this function will be performed at the time the first user accesses each page.

Make your selection, and then click **Next**.

A screen is displayed in which you can verify the information you have entered.

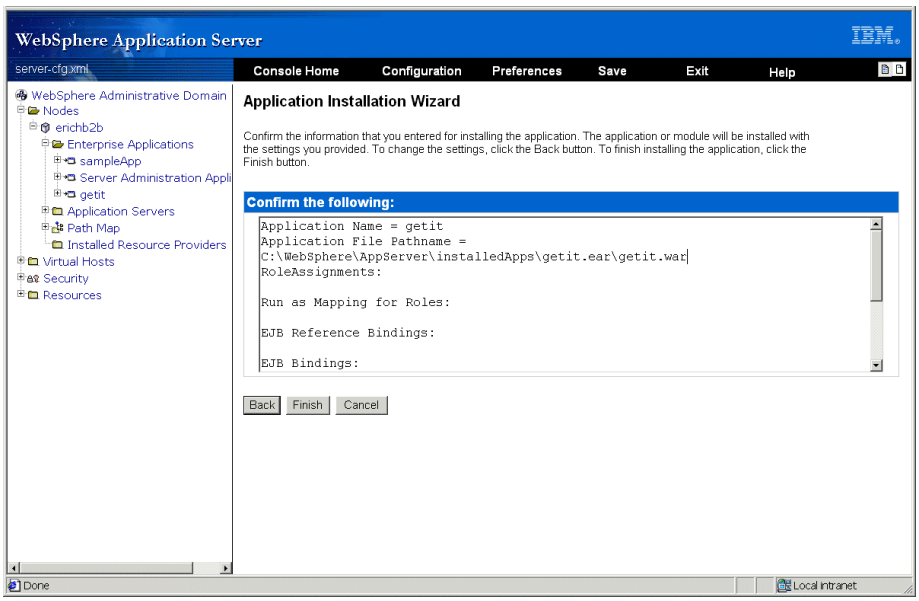

Fig. 3.4 Application Installation Wizard—confirm and set the pathname

9. Edit the Application File Pathname to be the correct path to the **getit.war** directory on WebSphere. Click **Finish**.

   The Wizard will now install the application. This can take 30 minutes or more to complete if you are precompiling the JSPs.

   When the installation is finished, a screen is displayed with three links at the top of the page.

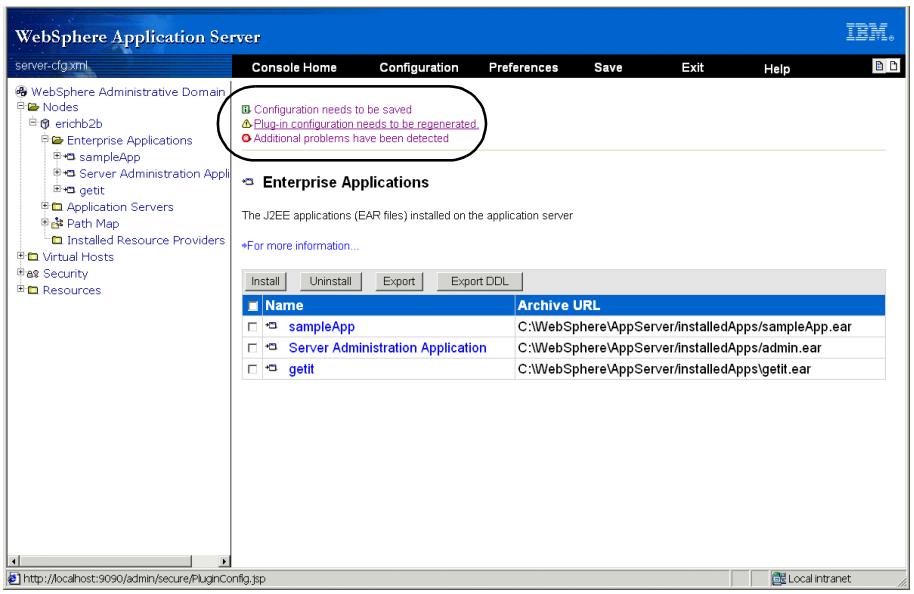

Fig. 3.5 Regenerate Plug-in configuration

10. Click “Plug-in configuration needs to be regenerated.”

    The Plug-in Configuration screen is displayed.

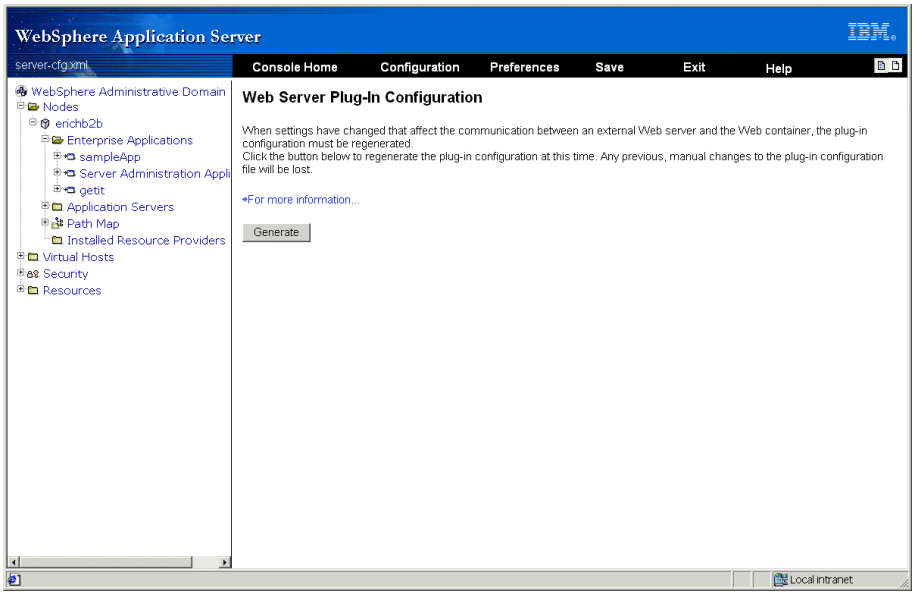

Fig. 3.6 Web Server Plug-In Configuration

11. Click **Generate**.

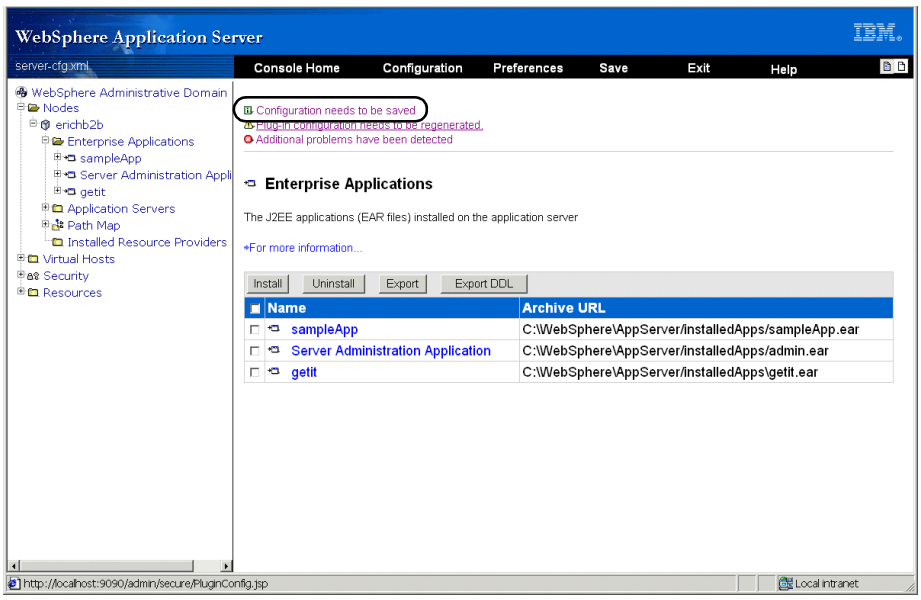

Fig. 3.7 Save the configuration

12. At the top of the screen, click “Configuration needs to be saved.”

    The Exit screen is displayed.

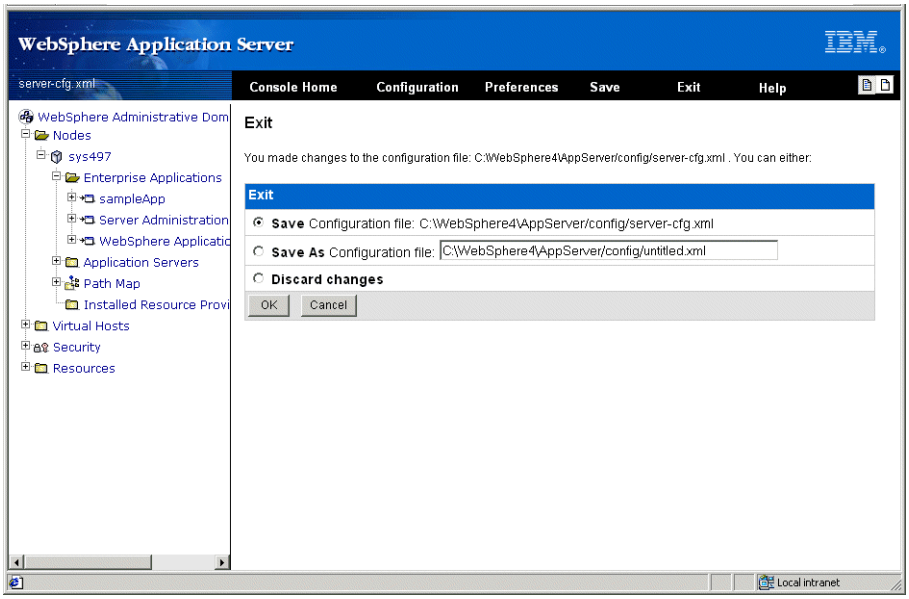

Fig. 3.8 Save and exit the Application Installation Wizard

13. Select **Save**, and then click **OK**.
14. Do one of the following:
    - If your WebSphere installation is on the same machine as your Get.It! installation, change the properties of the getit virtual directory to point to the getit\presentation directory. The Get.It! installer creates a virtual directory in IIS for use with JRun. This must be changed to use the getit.war\presentation directory under WebSphere\appserver\installedApps.
    - If your WebSphere installation is on a different machine than your Get.It! installation, create a virtual directory pointing to the getit.war\presentation directory.
15. Open a command prompt and in the WebSphere\AppServer\bin directory, type `stopserver` to stop the WebSphere server.
16. Go to the Get.It! lib folder (typically WebSphere\AppServer\installedApps\getit.ear\getit.war\WEB-INF\lib) and do the following:
    a. Move the **sunjce_provider.jar** file to the JRE extension folder (typically WebSphere\AppServer\java\jre\lib\ext).

b. If the **ibmjcefw.jar** file is in the JRE extension folder, remove the following files from the Get.It! lib folder:

   jce1_2_1.jar
   local_policy.jar
   US_export_policy.jar

   If the **ibmjcefw.jar** file is *not* in the JRE extension folder, move the above files from the Get.It! lib folder to the JRE extension folder.

17. In the WebSphere\AppServer\bin directory, type `startserver` to restart the server.

Reinstalling Get.It! on WebSphere

If you need to reinstall Get.It! on WebSphere, first uninstall your current installation using the WebSphere administrator's console as follows:

1. Open the WebSphere administrator's console (**Start>Programs>IBM WebSphere>Application Server V4.0**).
2. At the Login screen, click **Submit**. A user name is not necessary.
3. On the menu at the left side of the console, click **Nodes**, **<SystemName>**, and **Enterprise Applications**.

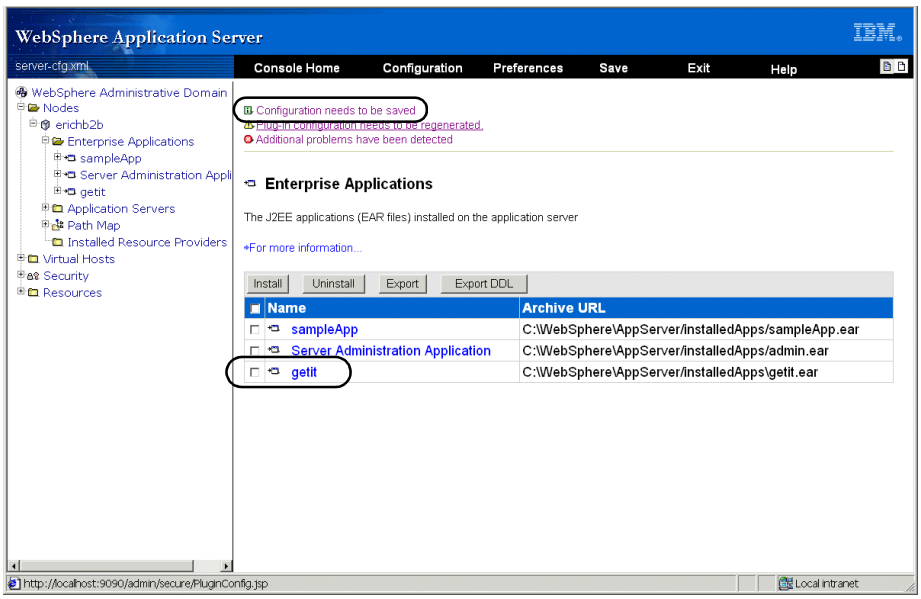

Fig. 3.9 Administrator's console—uninstalling Get.It! from the server

4. Select **getit**, and then click **Uninstall**.
5. At the top of the screen, click "Configuration needs to be saved."

The Exit screen is displayed.

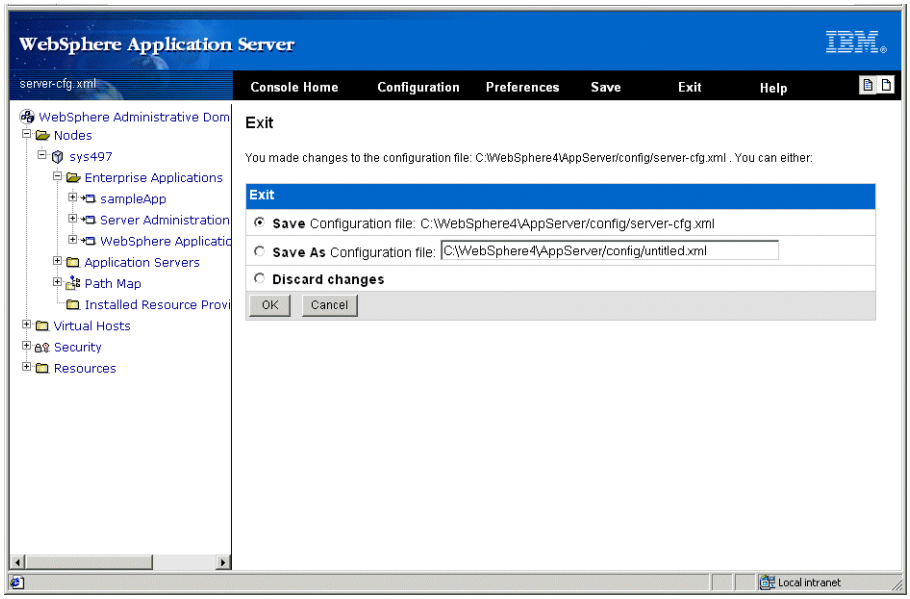

Fig. 3.10 Save and exit the Application Installation Wizard

6. Select **Save**, and then click **OK**.
7. Exit out of the console, and then stop and restart WebSphere. Repeat the configuration instructions to reinstall Get.It! on WebSphere.

Updating the scriptpoller.ini Files

If you will be using scriptpollers, you will need to update the scriptpoller INI files with the name of the Java virtual machine (JVM) for the application server you are using.

The **scriptpoller.ini** files are configured to use the default Java virtual machine (JVM). If you are using an alternate application server, you will need to tell Archway where to run the scriptpollers. The files are located in the getit\apps\b2b and getit\apps\common directories.

In the **scriptpoller.ini** file, add the ArchwayJVMName parameter, as shown in bold type in the example below, substituting `jvm_name` with the appropriate name for the application server you are using:

- WebSphere: getit_bin
- WebLogic: getit_wlserver60 (This will be different if WebLogic is not installed in the default installation directory.)
- Tomcat: getit_bin

```
<poller>
    <name>KeepAliveSC</name>
    <interval>600</interval>
    <parms>
      <ArchwayJVMName>jvm_name</ArchwayJVMName>
    </parms>
</poller>
```

Testing the Installation

To test that your system is properly configured for Get.It!, open your Web browser and type the following:

```
<server name>/getit/admin.jsp
```

This will display the login page for the Admin module in Get.It!. On the Control Panel, verify that the adapters are connected. For instructions for using the Admin module, refer to Chapter 5, "Get.It! Administration."

Chapter 4
Get.It! Architecture Overview

This chapter introduces the architecture behind Get.It!. Get.It! offers a simple and extensible way of using weblications to interface with Peregrine's existing systems, including AssetCenter and ServiceCenter, or with any other database.

Get.It! Architecture

Many of the examples shown in this chapter use AssetCenter and ServiceCenter specific file names and data, but you can adapt the information for use with any system.

Get.It! applications and interfaces are implemented using basic building blocks that include:

HTTP	A simple and widely supported protocol for sending client requests to a server. Variations such as HTTPS provide security as well.
XML	Extensible Markup Language. A documentation meta-language that allows you to format data, which can then be displayed through a Web browser. Unlike HTML, you create your own XML tags and define them any way you want.
Commercial web servers	The services provided by the Archway architecture can be served from any commercial Web server, including IIS, Apache, Netscape Enterprise Server, or the Java Web Server.
Application servers	Get.It! supplies JRun as an application server with the installation. Get.It! 2.0.1 also supports WebSphere, WebLogic, and Tomcat application servers.
Common clients	Applications can be built to be deployed via Web browsers (IE, Netscape), handheld devices (Palm Pilot), or mobile phones (through HDML).

The application server processes data (JSP pages, XML, and so forth) that it receives from the database or client that is specifically related to the Get.It! weblications. The Web server converts the data into a form (HTML) that can be displayed in a Web browser.

The following diagram illustrates the architecture:

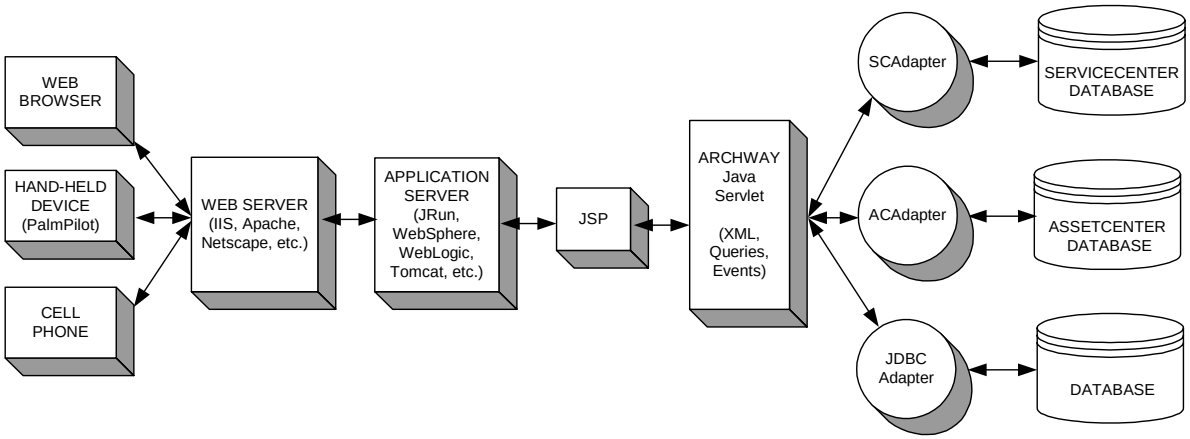

Fig. 4.1 Get.It! architecture

The Archway component listens to HTTP requests from clients, routes the requests to an appropriate server, and returns data or documents. The requests supported by Archway can vary, but they fundamentally consist of queries, data updates, or system events.

For example, a client can contact Archway and ask to query a database for a list of problem tickets. Another client could contact Archway and supply it with a new purchase request to be entered into the database.

All requests and responses are formatted using XML. For example, a problem ticket expressed in XML could appear as follows:

```
<problem>
  <number> PM5670 </number>
  <contact> Joe Smith </contact>
  <description> My printer is out of paper </description>
</problem>
```

Clients that interact with Archway can do anything they need with the XML that is returned as a response. Very frequently, the client initiating the request is a user interface such as a Web browser. Such a client could easily display the XML documents returned by Archway. However, to be of better use, the XML documents are often displayed within a formatted HTML page. This is accomplished by using Java Server Pages (JSP).

JSP provides a syntax for creating HTML pages that is pre-processed by the Web server before being sent to the browser. During this processing, XML data obtained from Archway is merged into the HTML page.

Archway's architecture includes special support for automatically generating the HTML and JSP pages that make up a weblication.

Archway Internal Architecture

Archway is implemented as a Java servlet. The Java servlet is an application executed by a Web server that processes HTTP requests from a client through a Web browser and sends the request, by way of an adapter, to a database. It then retrieves the requested information from the database and returns it to the client. Archway requires both a Java environment (JRun) and a Web server.

Each request is interpreted to determine its destination. Archway is able to communicate with a variety of back-end systems, including the AssetCenter or ServiceCenter products from Peregrine. Requests can be handled in one of three ways:

1. A request can be sent directly to an adapter that talks to a back-end server. For instance, a query request for opened tickets could be forwarded to an adapter capable of communicating with ServiceCenter.
2. A request can be sent to a script interpreter hosted by Archway. This enables you to define your own application-specific services. Within a script, calls can be made back to Archway to access the back-end system with database operations and events.
3. Finally, a request can be sent to a component known as a Document Manager. This component provides automated services for combining logical documents.

Archway communicates with back-end systems with the help of specialized adapters that support a predefined set of interfaces for performing connections, database operations, events, and authentication. All adapters use DLLs to communicate with each application.

Messages can be routed to a script interpreter hosted by Archway. The interpreter supports ECMAScript, a European standard based on the Core JavaScript language used by Netscape (JavaScript) and Microsoft Internet Explorer (JScript).

Messages can be routed to the Document Manager component. This component reads special schema definitions that describe application documents for logical entities such as a purchase request, problem ticket, or product catalog. The script interpreter uses these schemas to automatically generate database operations that query, insert, or update such documents.

Each form displayed by a weblication using Get.It! has a related JSP. The getit virtual directory tells the URL the location of the JSP pages the Web browser will use to display the weblication forms.

Archway Requests

Archway supports a variety of requests, all of which are based on two basic technologies: HTTP and XML. The HTTP protocol defines a simple way for clients to request data from a server. The requests are stateless and a client/server connection is maintained only during the duration of the request. All this brings several advantages to Archway, including the ability to support a large number of requests with the help of any of today's commercial Web servers.

Another important advantage is that any system capable of making HTTP requests can contact Archway. This includes Web browsers, of course. But in addition, all modern programming environments support HTTP. This makes it very simple to write new adapters that communicate with Peregrine servers without the need of specialized APIs.

An HTTP connection consists of:

- A client request
- A server response

The messages exchanged normally have a number of header lines and some content lines. For example, consider the following two principal parts of a request:

Query String	The parameters sent with the URL for the HTTP connection. For example: `http://prgn/servlet/archway?hello&world` This URL is made up of a server locator (`http://prgn/servlet/archway`) and a query string (`hello&world`).
Content	The data appended to the request. This data can be in any format, but for Archway, the data is always formatted as XML.

Archway uses the query string of a request to determine what it has been asked to do. The following query string syntax is expected:

```
archway?target.command&param=value&param=value&…
```

Let's consider each part of the request:

Target	The name of the target object that should handle the request. Archway forwards requests to a system and returns the response. Thus, the target could be ServiceCenter, AssetCenter, or another database. The target may also be the name of a Script Object that contains customizable logic for handling the request.
Command	The action that the target object should take. By default, five basic actions are supported: query, update, insert, delete, and event. However, when the target is a Script Object, the action can be any function defined by the script.
Param=Value	Parameter values included in the request. An arbitrary number of parameters can be passed along with the request. The encoding of these parameters is the same as that used by CGI (Common Gateway Interface). As with CGI, data sent by a browser is provided by fields embedded in an HTML form. This data is automatically formatted as a CGI request in a way that Archway understands.

The following are sample URLs that query Archway with HTTP requests. These queries return data in XML documents.

- `archway?sc.query&_table=probsummary&priority.code=1`

  This sends a query request to ServiceCenter for all records in the `probsummary` table with a priority code of 1.

- archway?ac.query&_table=amAsset&_return=Brand;mPrice;Model&_count=2

  This sends a query request to AssetCenter for the first two records in the **amProduct** table. Only the Brand, mPrice, and Model fields are returned for each record.

- archway?test.helloWorld&greeting=Hello

  This sends a *helloWorld* request to a script object named *test*.

Figure 4.2 illustrates shows the XML results of a query for products from AssetCenter.

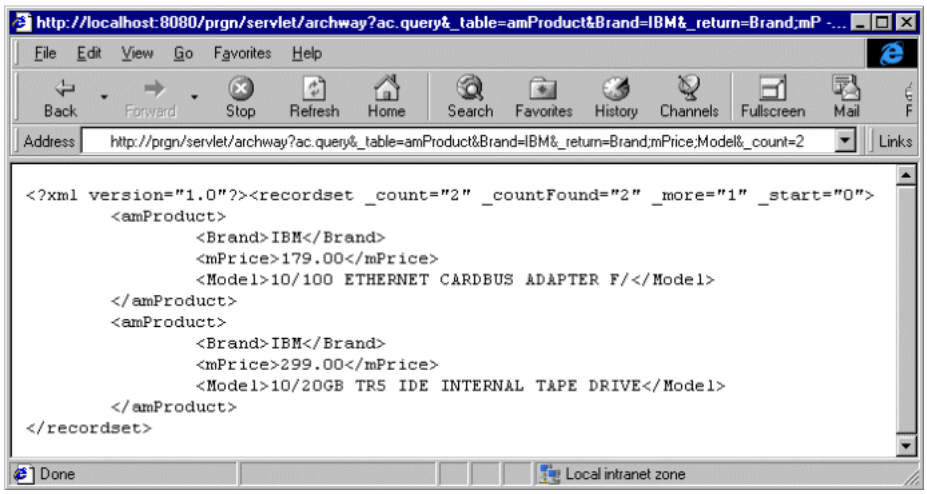

Fig. 4.2 Testing URLs from a Web browser

The Document Manager

Archway uses XML to exchange data and documents between clients and the supported back-end systems. Fundamentally, the XML data returned by Archway is obtained by executing queries against one or more systems. The queries can be executed by a direct URL request or indirectly within an ECMAScript.

Simple queries are only capable of returning record sets of data. However, clients are more often interested in exchanging documents. A Document is a logical entity built up of several pieces of data that can come from various physical database sources.

The Document Manager uses schemas to determine which XML elements to use and what data should be contained in the elements. The data used by the Document Manager depends on the back-end system being used (AssetCenter, ServiceCenter, or another form of database.

The *Get.It! Tailoring Guide* contains detailed information about schemas and the Document Manager.

Chapter 5
Get.It! Administration

Using the Admin Module

Get.It! includes an Admin module, which you can use to monitor how Get.It! is working and to change **archway.ini** parameters.

The Get.It! Admin module enables you to:

- Monitor the connection between the Get.It! server and the back-end servers.
- View the server log, which shows all activity on the Get.It! server.
- View and change settings in the **archway.ini** file.
- Start and stop various ECMAScript functions using Show Script Status.
- Show Message Queuing to view queue contents.
- Use Queue Status to view all queues.
- Change the adapter you use for the connection to your database.

Accessing the Admin Module

You can access the Admin module directly from a Web browser or from the home portal in Get.It!. You must log in as an administrator—a user with *getit.admin* as your user rights—in order to access the Admin module.

To access the Admin module from Get.It!:

1. Log in to Get.It! as a user with administrator rights.
2. Click the **Administration** button on the Home portal.

To access the Admin module from a Web browser:

1. Enter the following URL in the browser's address field:

   ```
   http://localhost/getit./admin.jsp
   ```

   Where localhost is the name of the server.
2. Press Enter.

The first time you access the Admin module from a Web browser through **admin.jsp**, you are asked to set up the administrator's ID and password because you are not using the login authentication process. Enter the information and click **Login as Administrator**. If you do not want to register at this time, you can click **Login as Administrator** without entering any information.

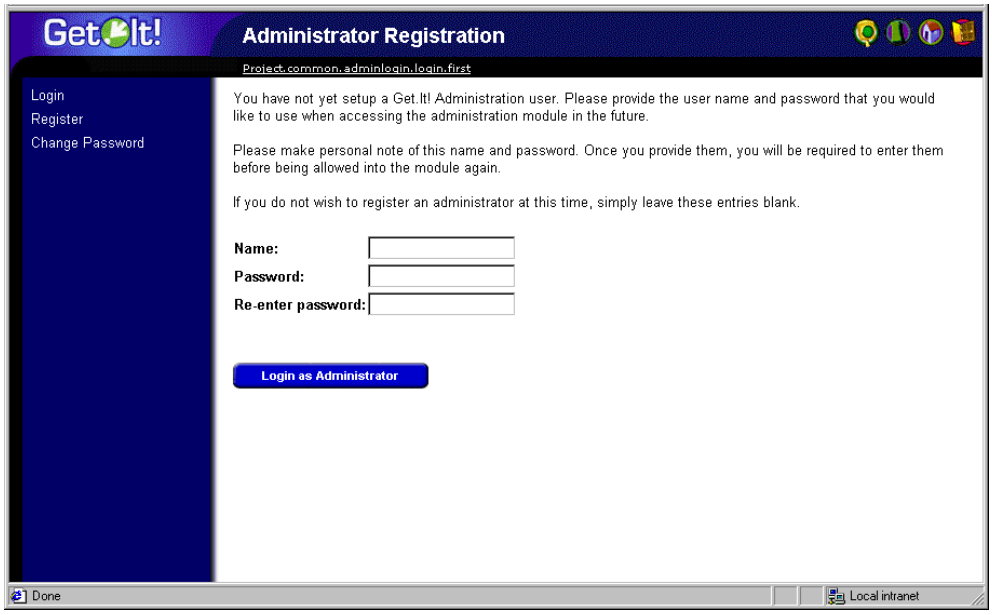

Fig. 5.1 Administrator Registration Form

The activities available in the Admin module are:

- Control Panel—check the status of connections to the back-end systems.
- Server Log—view activity on the Get.It! server.
- Settings—view and change settings in the **archway.ini** file.
- Show Script Status—verify which scripts are running. You can also start and stop scripts from this window.
- Show Message Queues—display a list of all message queues. Click on the queue name to see the contents of the queue.
- Show Queue Status—see the current status of the queues: operational and unlocked, or suspended. Click **Toggle Queue Operations** to change the status of the queues.

Using the Control Panel

When you first access the Admin module, the Control Panel is displayed.

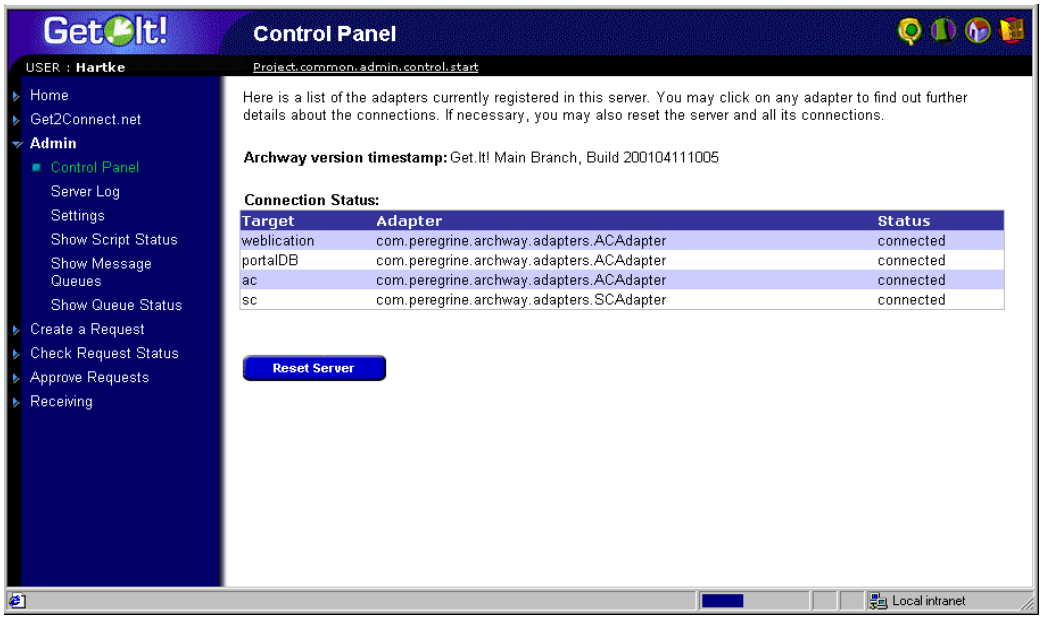

Fig. 5.2 Admin Module—Control Panel

Use this page to check the status of the connections to the databases you are accessing with Get.It!. The default connection for the weblication and portalDB is AssetCenter (ACAdapter). You can change the portalDB target to be ServiceCenter or use a JDBC connection to another database as your portalDB setting. These settings are changed using the Admin module Settings screen, described later in this section.

You can also reset the connection between the Archway servlet and the adapters to the back-end systems:

1. Click **Reset Server**.
2. A message is displayed at the top of the page indicating that the connections have been reset.

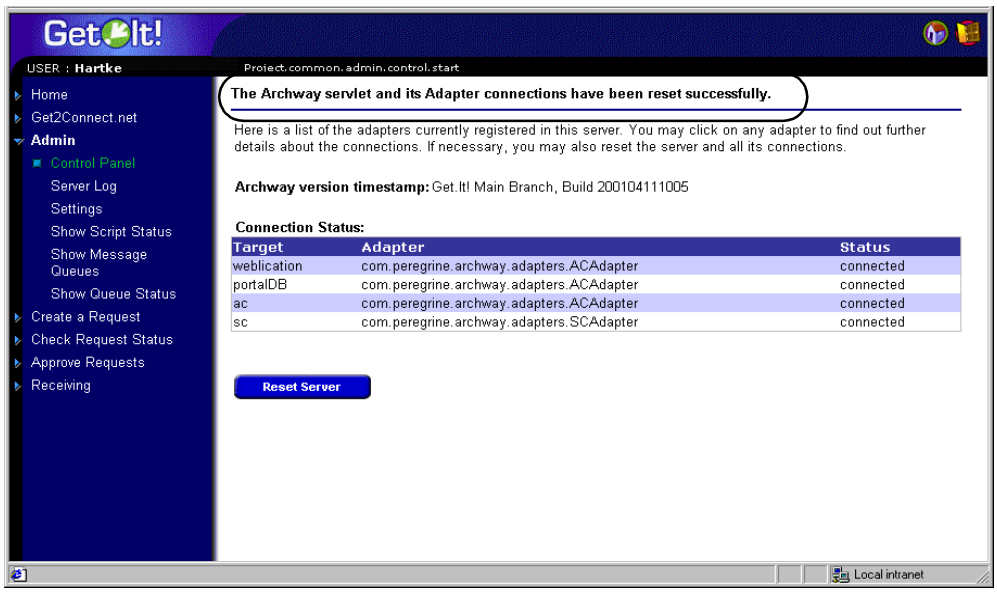

Fig. 5.3 Admin Module—Servers Reset

Viewing the Server Log

The Server Log provides a history of server events.

To view the server log:

1. Select Server Log from the sidebar activities menu.

   A form is displayed from which you can select the log you want to view. You can select the default **archway.log** file or a backup log file.

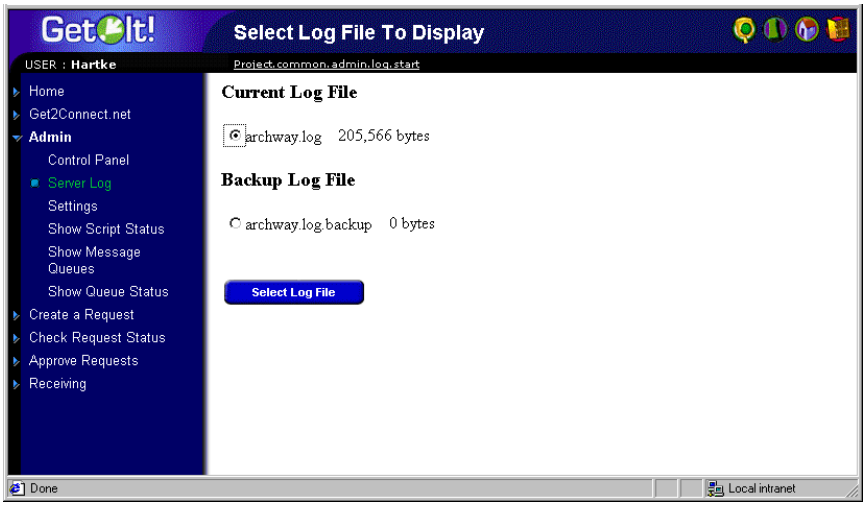

Fig. 5.4 Admin Module—Server log selection

1. Select the log file you want to view.
2. Click **Select Log File**.

   A search form is displayed. You can search for a particular item, search from a particular time to the end of the log, and set the number of lines of the log entry that you want to view.

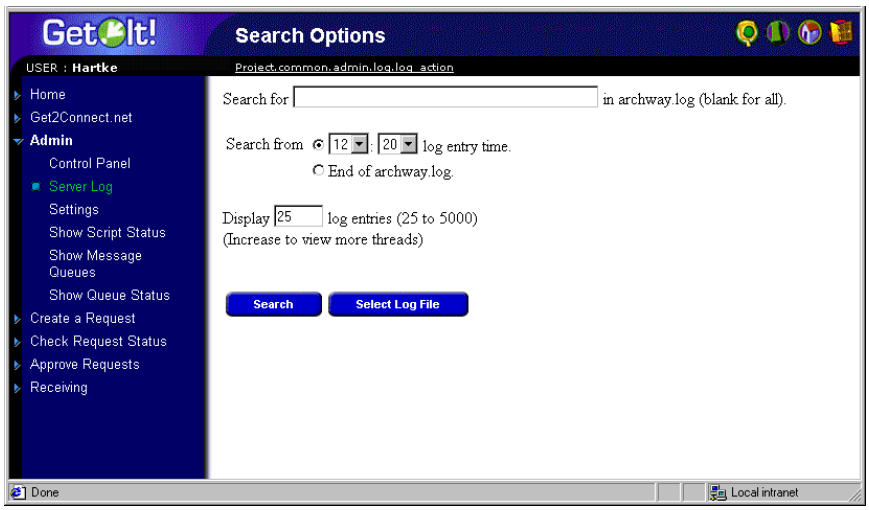

Fig. 5.5 Server Log search options

3. When you have made your selections, click **Search**.

The log file entries you have requested are displayed.

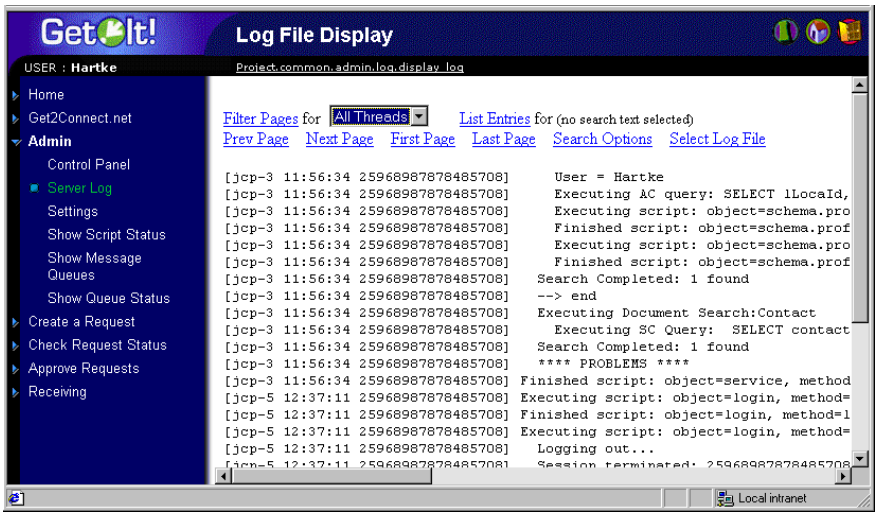

Fig. 5.6 Server Log display

Use the links at the top and bottom of the log file display to navigate through the log. You can also return to the Search Options or Select Log File pages to make new selections. The **Filter Pages for** drop-down list lets you select particular lines to view.

Using the Settings Page

Click **Settings** to display the current settings of parameters in the **archway.ini** file. The Settings page is divided into tabs. The tabs you see will depend on which weblications you have installed. The Common tab is available for all installations.

- Common—includes settings that are common to all weblications.
- Get.Resources!—parameters that are specific to the Get.Resources! weblication.
- Get.Service!—parameters that are specific to the Get.Service! weblication.

This chapter includes information about the settings on the Common tab. Settings for Get.Resources! and Get.Service! are included in the Quick Start Guides for these weblications.

This section also includes a table of parameters that can be set directly in the **archway.ini** file.

Each available option is explained on the Settings page, as shown in Figure 5.7.

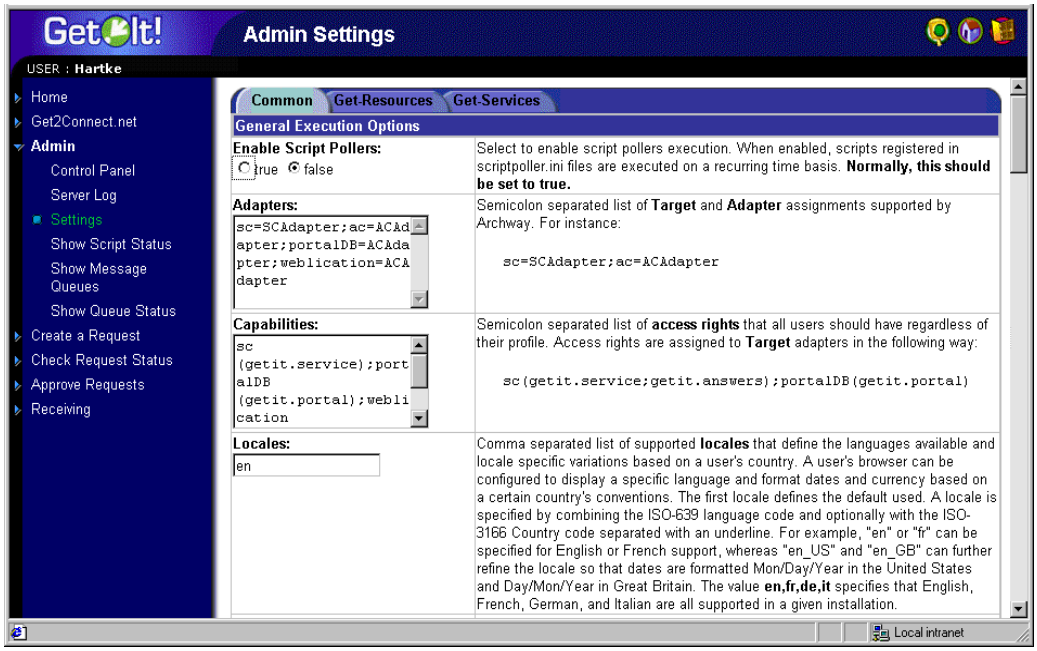

Fig. 5.7 Admin Module—Settings

Setting Parameters using the Admin Module

The following tables list the parameters on the Common tab. The default values are also shown, along with a description of how the parameter is used.

Note: Some of these parameters are specific to AssetCenter or ServiceCenter and do not apply if you are not using these products.

General Execution Options

Admin Module Field	Default Setting	Description
Enable Script Processing	false	Select to enable script processing. Normally, this should be set to *true*. A setting of false will stop your server from performing background tasks.
Adapters	adapters=sc=SCAdapter; ac=ACAdapter; portalDB=ACAdapter; weblication=ACAdapter	Semicolon-separated list of target and adapter assignments supported by Archway.
Capabilities	portalDB (getit.portal)	Semicolon-separated list of access rights that all users should have regardless of their profile. Access rights are assigned to target adapters in the following way: portalDB(getit.portal)
Locales	en	Comma-separated list of supported locales that define the languages available and locale specific variations based on a user's country. A user's browser can be configured to display a specific language and format dates and currency based on a certain country's conventions. The first locale defines the default used. A locale is defined by combining the ISO-639 language code and (optionally) with the ISO-3166 Country code separated with an underline. For example, *en* or *fr* can be specified for English or French support, whereas *en_US* and *en_GB* can further refine the locale so that dates are formatted Mon/Day/Year in the United States and Day/Mon/Year in Great Britain. The value *en,fr,de,it* specifies that English, French, German, and Italian are all supported in a given installation.
Currencies	USD	Comma-separated list of ISO-4217 currency codes that define which currencies will be used in a given installation. The first code defines the default currency to use when a currency type is not specified. For example, USD,EUR defines US Dollars and European Euros as supported currencies.

General Execution Options (Continued)

Admin Module Field	Default Setting	Description
Content Type Encoding	ISO-8859-1	The document type encoding used for generating documents used by the generated JSP pages. This value should match the content type value used in Get.It! Studio for generating the JSP pages.

Logging

Admin Module Field	Default Setting	Description
Debug Logging	false	Set this to *true* to generate log information useful when troubleshooting the server.
Log Domains	(none)	A semicolon-separated list of execution log traces you can enable. Choices include: dll - Adapter DLL loading and unloading. weblication - Weblication and personalization rendering. jvm - Java run-time environment management and status.
Log File	c:\getit\archway.log	The full directory path to the file used for logging.
Log Archive Zip File	c:\logs.zip	The full directory path to the Zip file used for daily archiving of the log. Leave blank to omit archiving.
Daily Log File Archive Time	00:00:01	The time of day to make the daily archive of the log to the zip file as HH:MM:SS or HH:MM. Leave blank to omit archiving.
Number of days to Archive	7	Number of days to save in the daily archive of the log to the zip file. Set to 0 to omit archiving.
Log Entries to View	25	The number of log entries you want displayed on each page.

Weblication Settings

Admin Module Field	Default Setting	Description
Cookie Expiration	2592000	Number of seconds to store weblication user settings as browser cookies. For instance, a user's name and password are remembered by a weblication by storing them in a cookie.
Show Form Info	false	When set to *true*, form information is displayed in each screen to aid during weblication development and customization.
Session Timeout	6000000	Number of seconds to allow an inactive session to exist before the user is automatically logged out.
Personalize Hierarchy	personalize.getHierarchy	Name of a script that returns a user personalization hierarchy. The default is to call **personalize.getHierarchy**. See the script for details.
Strip Authentication Domain	false	When set to *true*, the Domain name is stripped from the login name (e.g., DOMAIN\login, becomes login). This setting is used by NT Challenge/ Response for authentication and should be set to *false* when you are using AssetCenter 3.51 or later. However, if the AssetCenter and ServiceCenter user profiles have only the login name without the domain, set this parameter to *true*.
End User Personalization	limited	Get.Resources! and Get.Service!—The level of personalization access granted to end users: Enabled - End users have full access to personalize screens. Disabled - End users have no access to personalize screens. Limited - End users can only move or remove existing fields in a form.

Themes - Skins and Stylesheet Settings

Admin Module Field	Default Setting	Description
Default Logo	images/logos/ getit_header_logo.gif	Sets the global logo to be used in the application. The logo is skinned and is located at the root level of each skin directory in Themes. To add a new custom logo, add it to the skin template, using instructions in the *Get.It! Tailoring Guide*.
Default Stylesheet	classic.css	The default stylesheet that is used when a user first logs into Get.It!. If you change the default to another style, this style will be recorded in a database as well as in the browser's cookie.
Skins/Themes	skins/	Sets the Skins directory location. The directory name must be specified relative to the presentation directory. Setting this allows you to move the default location of the skins directory to another location.
Default Skin/Theme	classic	The default skin that is used when a user first logs into Get.It!. If you change the default to another style, this style will be recorded in a database as well as in the browser's cookie.

AssetCenter Adapter Settings

Admin Module Field	Default Setting	Description
Admin Name	Admin	Administration user name used by AssetCenter when performing authentication and registration.
Admin Password	(none)	Administration password used by Get.It! when performing tasks such as user authentication and registration in AssetCenter.
Anonymous Name	Admin	Anonymous user name for AssetCenter.
Anonymous Password	(none)	Anonymous user password for AssetCenter.

AssetCenter Adapter Settings (Continued)

Admin Module Field	Default Setting	Description
Database	(none)	The name of the AssetCenter database you have installed.

ServiceCenter Adapter Settings

Admin Module Field	Default Setting	Description
Admin Name	falcon	Administration user used by Get.It! when performing tasks such as user authentication and registration in ServiceCenter.
Admin Password	(none)	Anonymous user password for ServiceCenter.
Anonymous Name	falcon	Anonymous user name used when an unknown user attempts to communicate with ServiceCenter
Anonymous Password	(none)	Anonymous user password for ServiceCenter.
Host	localhost	Host name of the ServiceCenter server.
Port	12670	Port number of the ServiceCenter server.
Log	(none)	Path to SC logging used by the ServiceCenter client connection.

Email Settings

Admin Module Field	Default Setting	Description
Legal Domains	peregrine.com; apsydev.com; getmarketaccess.com	A semicolon-separated list of mail domains that Get.It! can correspond with. Only users with an e-mail address in these domains are allowed to complete online self-registration.
Mail Host	(none)	Mail server host used by Get.It! to send e-mail messages.
Mail User	(none)	User name used to access the mail server.

Email Settings (Continued)

Admin Module Field	Default Setting	Description
Mail Password	(none)	User password used to access the mail server.
Mail Sender	(none)	Reply address used when sending e-mail.

Advanced Settings

Admin Module Field	Default Setting	Description
Session Tracking	(none)	Specifies how session-related data is tracked: Memory - Fastest choice, but depends on continuous service from one Web server. Cookies - Maintains session independent from the Web server. Limits amount of data that a session can track. Database - Maintains session independent from the Web server. Less efficient, but also most reliable. (Currently requires ServiceCenter.)
Event Queue	(none)	The name of the adapter to be used by the Get.It! event queue engine. For example: To use ServiceCenter's repository, enter “sc”. To use AssetCenter's repository, enter “ac”.
Script Threads	10	Maximum number of script execution threads.
Script Timeout	0	Number of milliseconds to allow an inactive session to exist before autologout.
Garbage Collector Timer	(none)	Activates Java memory garbage collection based on a timer of N seconds. The default is to perform garbage collection every 60 seconds. A setting of 0 disables the timer.

Setting Parameters in the archway.ini File

The following table lists additional parameters that can be set directly in the **archway.ini** file.

Parameter	Description
ie_css=css/	Directory path for CSS stylesheet directory for Internet Explorer.
images=images/	Sets the Skins directory location. The directory must be specified relative to the "presentation" directory. Setting this allows you to move the default location of the skins directory to another location.
maxlogsize=1000000	Maximum size for the log file in bytes.
acdefaultloginclass	Establishes the type of default user login. This parameter can have four different values: • If it is empty or not specified in **archway.ini**, the self-registered users are added to AssetCenter with the default login class defined in AssetCenter. The default for a registered user in AssetCenter is *named* user. • If the value is *casual* (acdefaultloginclass=casual), the self-registered user will be added as a casual user. • If the value is *floating* (acdefaultloginclass=floating), the self-registered user will be added as a floating user. • If the value is *named* (acdefaultloginclass=named), the self-registered user will be added as a named user. Casual user, floating user, and named user are described in the AssetCenter documentation.
ns_css=ns_css/	Directory path for CSS stylesheets directory for Netscape Navigator browsers.
acdateformat=yyyy-MM-dd	AssetCenter date format.

Parameter	Description
debugscript=true	When enabled, scripts and schemas are reparsed each time they are invoked. Be sure to turn this off in a production system.
jrunroot=C:\JRun	The default root directory for the JRun installation.
appspath=apps\	Directory location of the Get.It! weblications.
<target_name>.AdapterPoolSize=n	Defines resource pooling for the JDBC Adapter, E-mail Adapter, and LDAP Adapter. The default setting is 1.

Verifying Script Status

The name and status of any script that is currently running is displayed on the Script Status page. Click on the script to enable or disable it.

Displaying Message Queues

The Message Queues are displayed whenever a queue has data waiting to be transferred. This activity is used primarily by B2B.

Showing Queue Status

To verify or change the status of the message queues, select **Show Queue Status** on the activity menu in the sidebar. Click **Toggle Queue Operations** to change the queues from *operational and unlocked* to *suspended*.

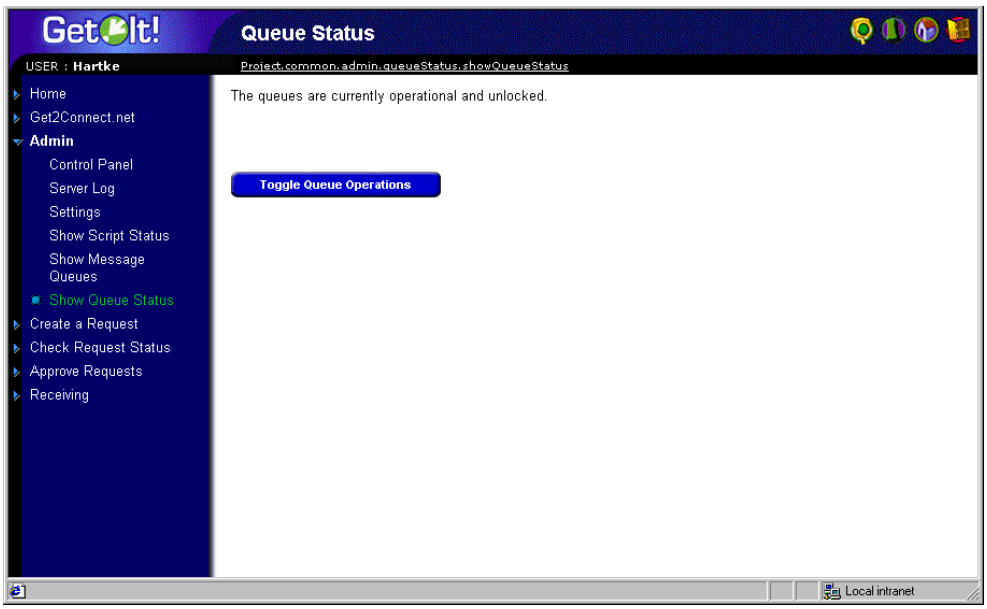

Fig. 5.8 Admin Module—Queue Status

Displaying Form Information

You can use the Admin module to configure Get.It! forms to display the location and file name of the current form.

To display form information:

1. With the Admin module displayed, click Settings.
2. Scroll down to the Weblication Settings section.
3. In the Show Form Info field, select the *true* check box.

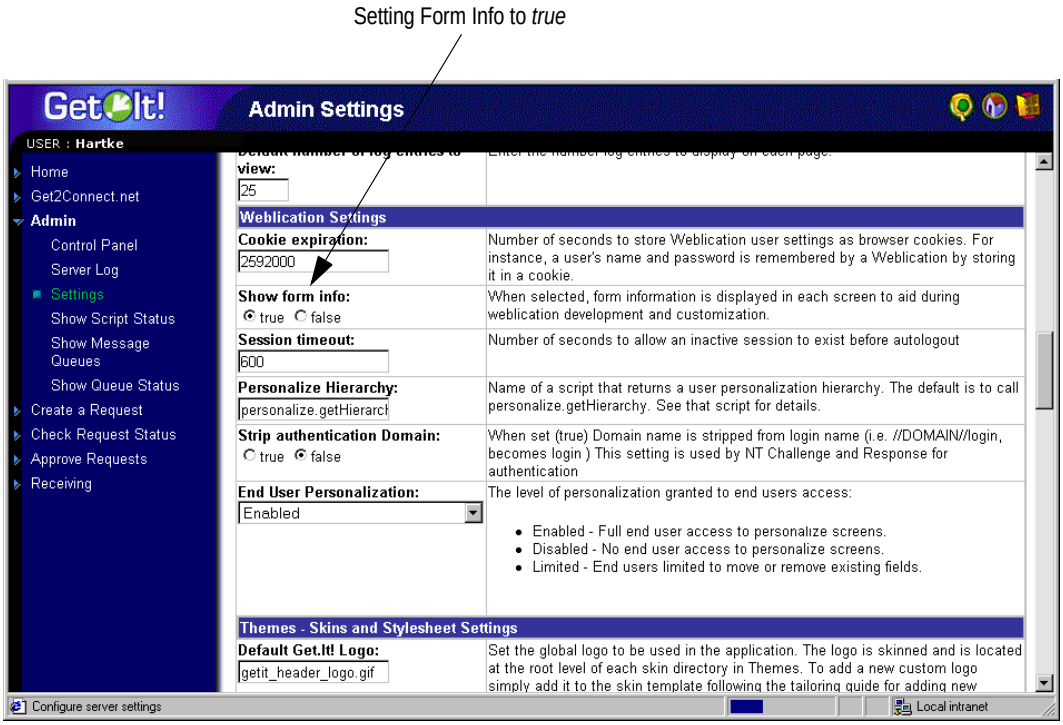

Fig. 5.9 Admin Settings—Show Form Info

4. Scroll to the bottom of the page and click **Save**.

   The name of the form is displayed at the top of each form in Get.It!, as shown in Figure 5.10.

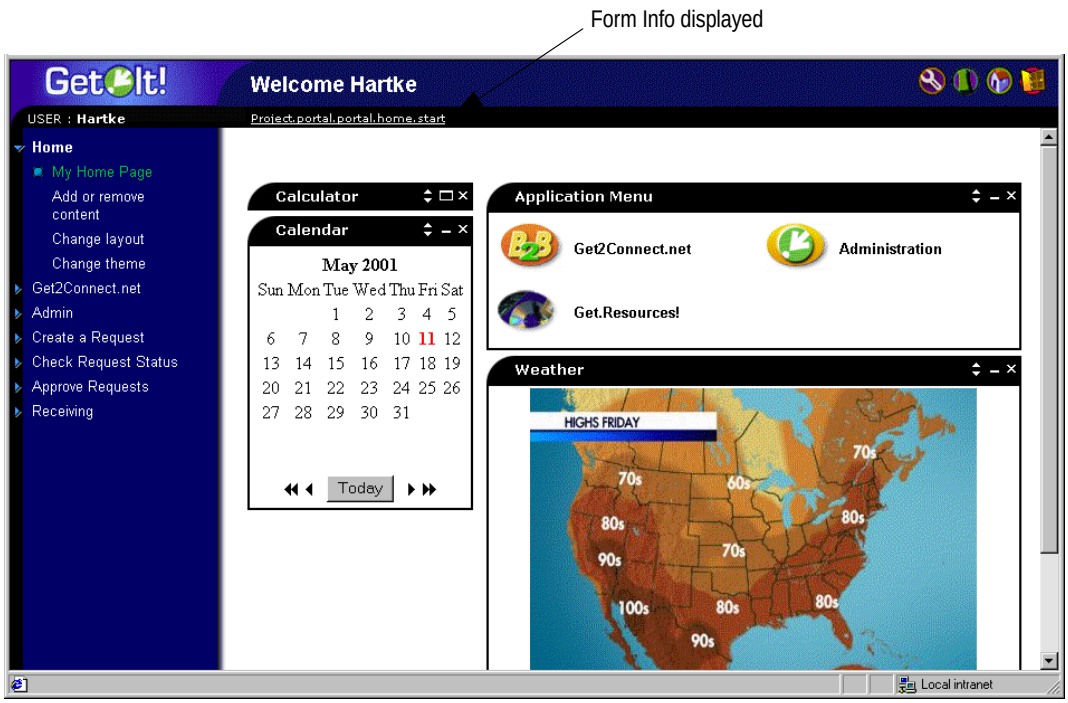

Fig. 5.10 Form Info displayed

Note: If you have Get.It! Studio open with the project loaded that contains the form you are viewing, you can click on the form information link and the form will be opened in Get.It! Studio for editing. Refer to the *Get.It! Tailoring Guide* for detailed instructions for using Get.It! Studio.

Displaying Form Details

You can also display detailed information about the current form. Click on the **Form Info** icon at the top right of the form. A separate window is displayed, with the following tabs:

- Script Input—the script that sends a request to the back-end system.

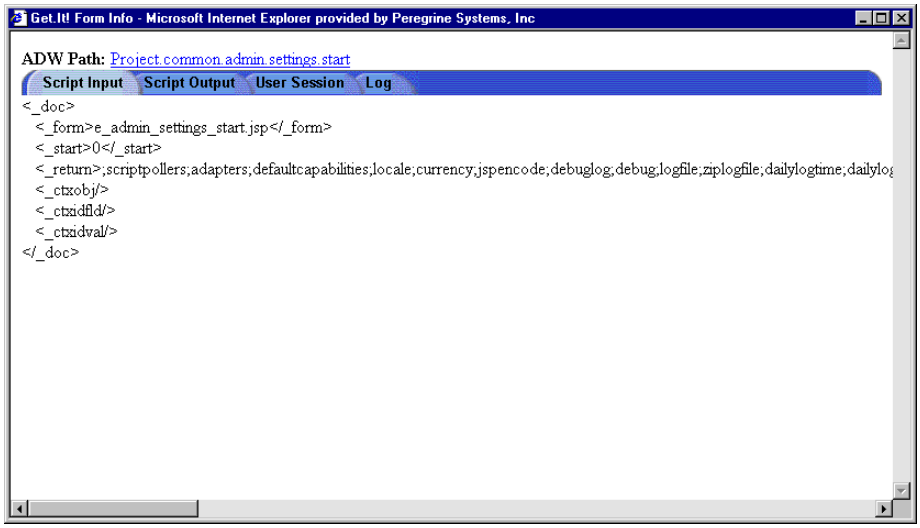

Fig. 5.11 Show Form Info—Script Input

- Script Output—the information returned by the script request to the back-end system.

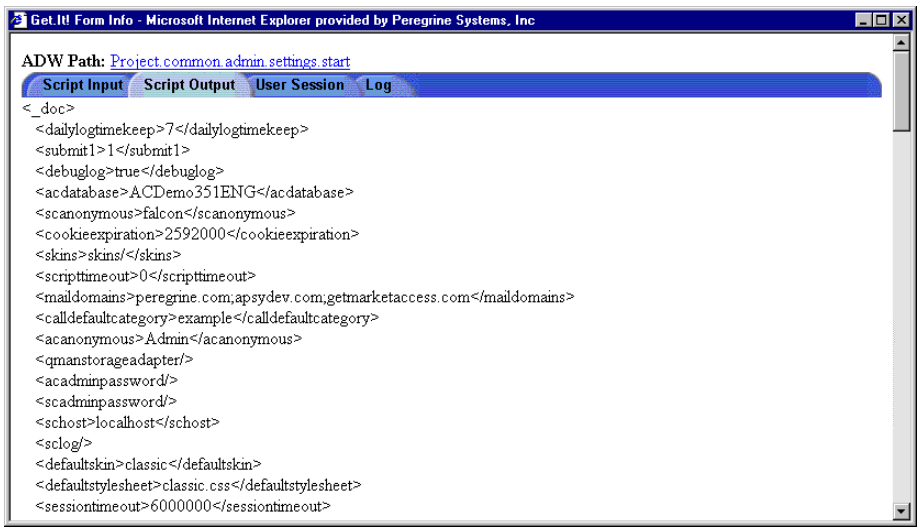

Fig. 5.12 Show Form Info—Script Output

- User Session—details about the current user session, including browser type, back-end system version, and the access rights established for this user.

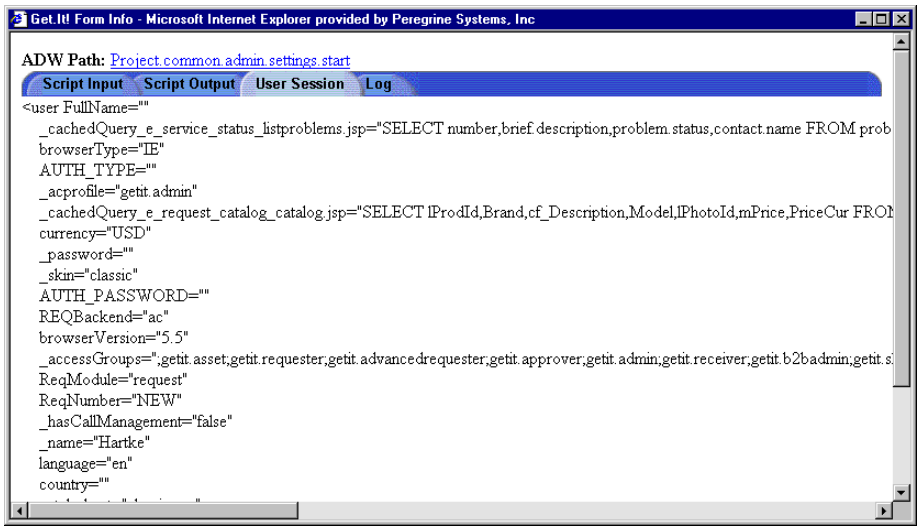

Fig. 5.13 Show Form Info—User Session

- Log—log of actions taken by the script to execute the form.

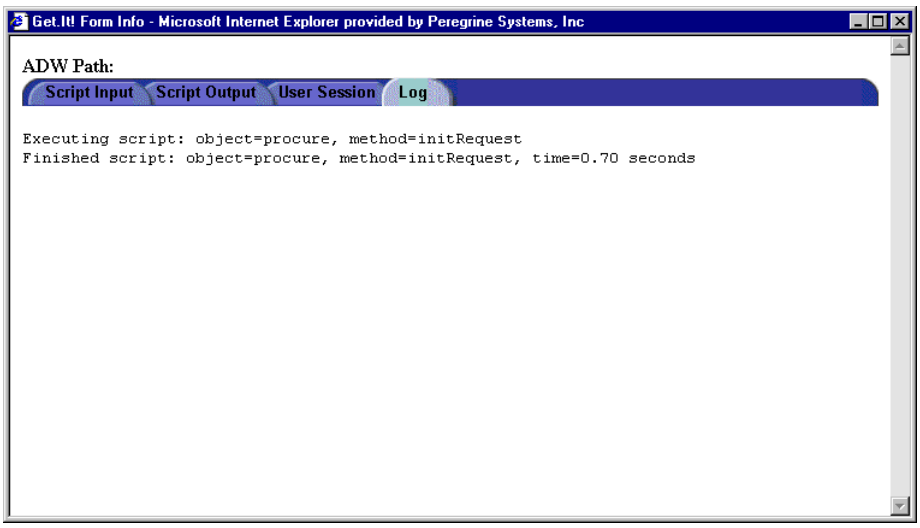

Fig. 5.14 Show Form Info—Log

Chapter 6
Security

This chapter includes information about Get.It! security, including the following:

- User registration
- User authentication
- Installing and using Windows NT Challenge/Response

Get.It! Security Overview

Get.It! operates transparently over Secure Sockets Layer (SSL) when the virtual directory of the Web server is configured to use SSL. No certificate-based authentication is used. Passwords are sent over the wire as plain text unless SSL is employed. Passwords are stored as plain text in a browser cookie if the user selects **Enable automatic login**.

SSL uses the public-and-private key encryption system, which includes the use of a digital certificate. If your Web server has a certificate, then your URL would be something like https://webserver/getit. Most browsers (Internet Explorer 4.0 or higher and Netscape 4.x) recognize this and know to send data encrypted when the site to which it is connecting is a secured site (https).

Get.It! also supports Windows NT Challenge/Response authentication. When this form of authentication is used, passwords are not actually exchanged between the browser and Web server, and the authentication process is kept secure. However, Windows NT Challenge/Response is only supported by Internet Explorer browsers on Windows systems. Instructions for configuring Windows NT Challenge/Response begin on page 6-6.

User Registration

If a user has a login name already established in the back-end system you are using with Get.It!, the user does not need to register in Get.It!. For example, in ServiceCenter the appropriate capability words would be defined in the user's Operator record. In AssetCenter, the appropriate user rights would be defined in the user's Profile. Similar access rights can be defined in whichever back-end system you are using. The user login will automatically be authenticated in the back-end system.

However, if a user is attempting to log in for the first time without back end authentication, the user is prompted for certain default information as shown in Figure 6.1. Note that the first four fields are required, as indicated by the arrows to the right of each field.

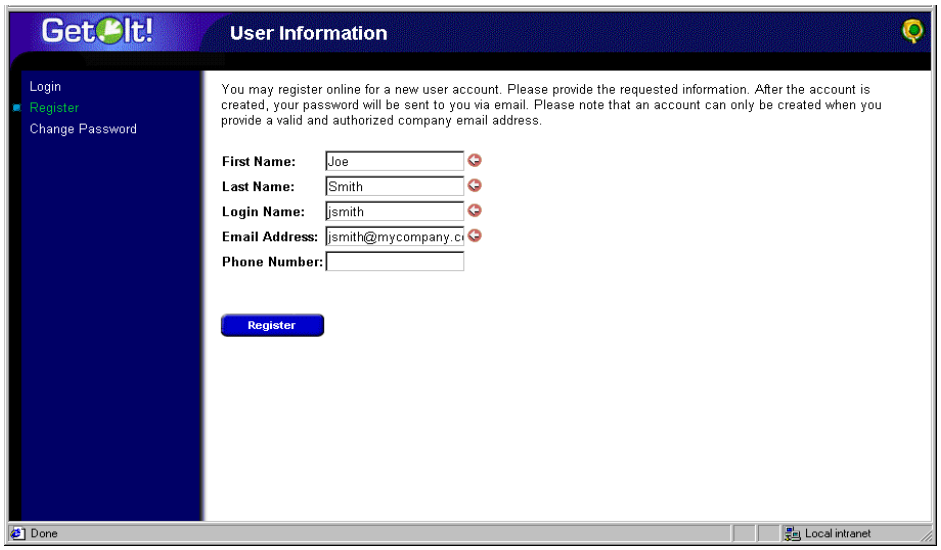

Fig. 6.1 Registering a new user

When the user clicks **Register**, the information is stored in the appropriate database. If you are using AssetCenter, Get.It! will transform this data into a Profile record that will then be passed to your AssetCenter system. An amEmplDept record is created with the user-supplied data and a default Profile will be assigned, *getit.default*.

If you are using ServiceCenter, an Operator record is created. A contact record is also created for the new user.

The adapter must be defined before the capability words will be recognized. For example, if no adapter is defined for Service Center, the ServiceCenter capability words will not be used.

Your AssetCenter and ServiceCenter documentation include instructions for establishing user rights and using capability words. The access right, getit.admin, can be defined for any user who will need administrative access. Other access rights, which are specific to the weblications, are explained in the Quick Start guides for the Peregrine weblications you have purchased.

Basic registration information and login scripts are stored in the .../getit/apps/common/jscript/ directory. Login scripts are in the file named **login.js**. If you want to make changes to the registration process, such as changing the way a user's password is defined, you can change the scripts in this directory.

User Authentication

All user authentication is provided through Archway. When a user attempts to log in to a weblication, the user name and password entered are validated against the back-end system you are using (for example, in AssetCenter, the user profile and in ServiceCenter, the operator record). The name and password combination may be valid in none or one or more systems. If the entered combination is invalid or does not exist in any systems, the user will be prompted to enter a valid user name and password. If correct in all systems, the weblication will retrieve the access rights for the user and log the user into Get.It!. If the combination is valid in one but not all systems, the access rights defined in the profile where the user name and the user password are valid will be used.

For example, if you are using AssetCenter and a user does not have access rights to a particular table in AssetCenter, the user will not be able to access the corresponding module in a Get.It! weblication. The same is true if you are using ServiceCenter for the back-end system. The user must have the appropriate capability words set in the operator record in ServiceCenter in order to see the corresponding module in a Get.It! weblication.

Note: When you are establishing passwords, make sure you use the Admin module. Passwords created from the Admin module are encrypted. Passwords entered directly in the **archway.ini** file are not encrypted.

Overriding the Login Script

You can override the login script function, login.init(), to obtain login credentials from another source. To accomplish this, you need to create a function that does two things:

- Obtains the user's login name.

  For example: `msg.set( "loginuser", strUser );`

- Turns on a flag that lets Get.It! know that the user has been pre-authenticated.

  For example: `user.set( "logintype", Archway.Archway.PRE_AUTHENTICATED;`

For an example of how information is passed to the Get.It! **login.jsp** file for authentication, look at **login.asp**, **logindefault.asp**, and **loginmgr.asp**. The data that is passed to **login.jsp** is parsed by the server side JavaScripts **login.init** and **login.login**.

Implementing this functionality requires that you use a different login screen for the initial Get.It! page. This screen does authentication, such as Windows NT Challenge/Response, and passes the authentication to the **login.jsp** file. The **login.jsp** file then sets up the initial frame set, and the server script functions, login.init() and login.login() are called.

The following diagram shows the process for authenticating a login within Get.It! as shipped.

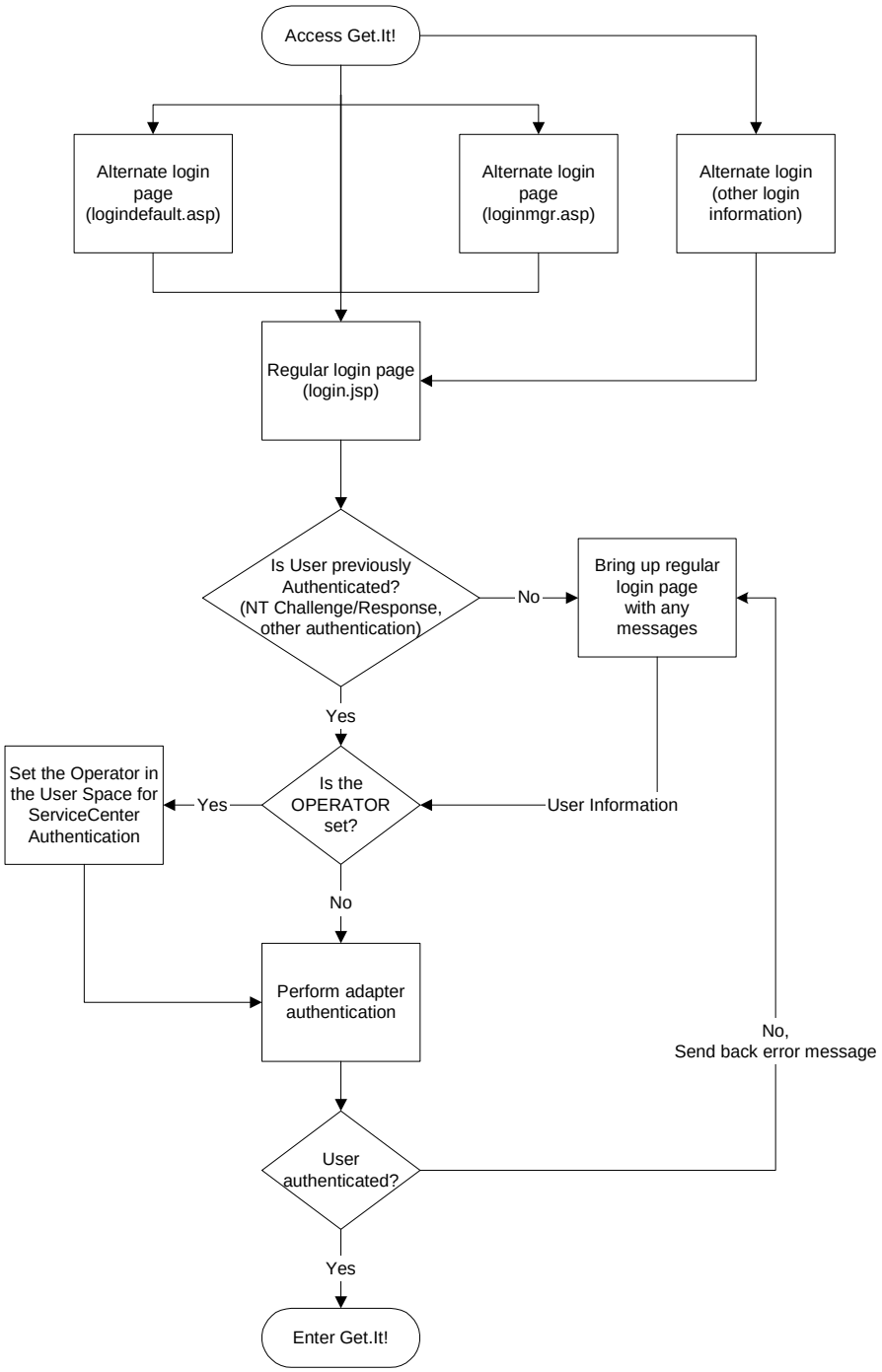

Fig. 6.2 User authentication at login

Windows NT Challenge/Response

Windows NT Challenge/Response is one of the ways NT facilitates the authentication of users on a Web server. The process consists of a secure handshake between the browser (IE) and the Web server (IIS). The handshake lets the Web server know exactly who the user is, based on how they logged in to their workstation. This allows the Web server to restrict access to files or applications based on who the user is. Applications running on the Web server can use this information to identify users without requiring them to log in.

Get.It! uses Windows NT Challenge/Response as follows:

- The user logs in to an NT workstation.
- The user starts an IE browser and navigates to the Get.It! **login.asp** page.
- IE automatically sends user authentication information to IIS. The user's password is not transferred, but the Windows NT Challenge/Response handshake between IE and IIS is enough for the server to recognize the user.
- The Get.It! login automatically detects the user by using the Windows NT Challenge/Response/IIS server data.
- Get.It! logs in the user without requiring that a name and password be entered.

During this process, Archway authenticates and impersonates the NT user with each of its adapters.

The following circumstances must be handled during this process:

- The NT user is not yet registered with an Archway adapter. When this occurs, Get.It! asks the user to register and enter profile information. Get.It! then lets the user log in and stores this information for future login attempts.
- The NT user name is already registered as an Administrator in the back-end system. When this occurs, Get.It! does not proceed with automatic login. The user is presented with another login screen and is asked to verify their password. This step is an added security measure to prevent a user from accidentally logging in with administrative rights.

Setting Up Windows NT Challenge/Response

The following procedure uses Windows NT as an example.

If you are using a system using Windows 2000, the overall procedure is the same. However, in Windows 2000, Challenge/Response is called Integrated Windows Authentication, and the IIS Management Console is called Internet Information Services.

1. Open the IIS Management Console (**Start>Programs>Administrative Tools>Internet Services Manager**).
2. Click on the getit virtual directory.
3. Right-click on **login.asp** and select **Properties**.
4. Select the File Security tab.
5. Click **Edit** in the "Anonymous Access and Authentication Control" section.

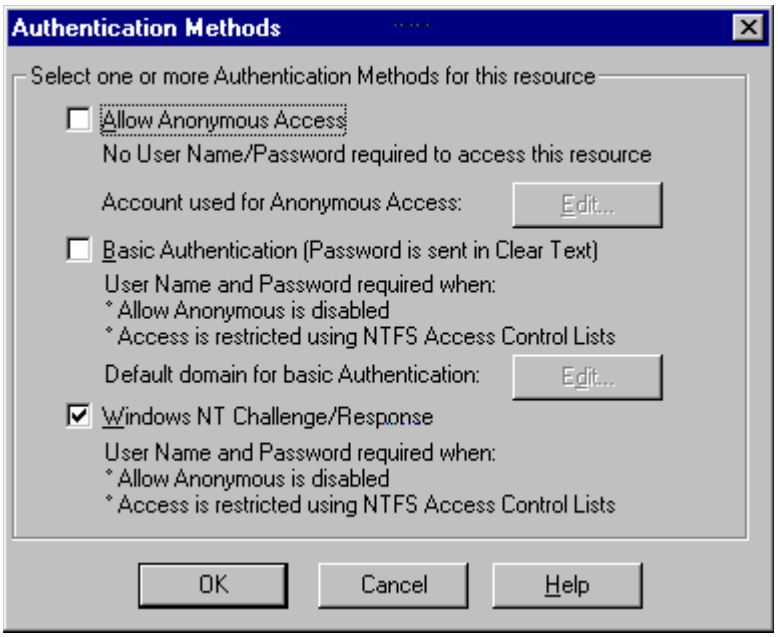

Fig. 6.3 Setting Authentication for login.asp

6. Check **Windows NT Challenge/Response**. Make sure this is the only option checked. Click **OK**.
7. Click **OK** on the other windows displayed until you return to the Microsoft Management Console.

Updating the loginverify.asp File

1. Repeat the steps above for **loginverify.asp**. Follow steps 3 through 5 as they are written above except select **loginverify.asp** instead of **login.asp**.
2. In the Authentication Methods window, check the **Allow Anonymous Access** and **Windows NT Challenge/Response** options. Click **OK**.

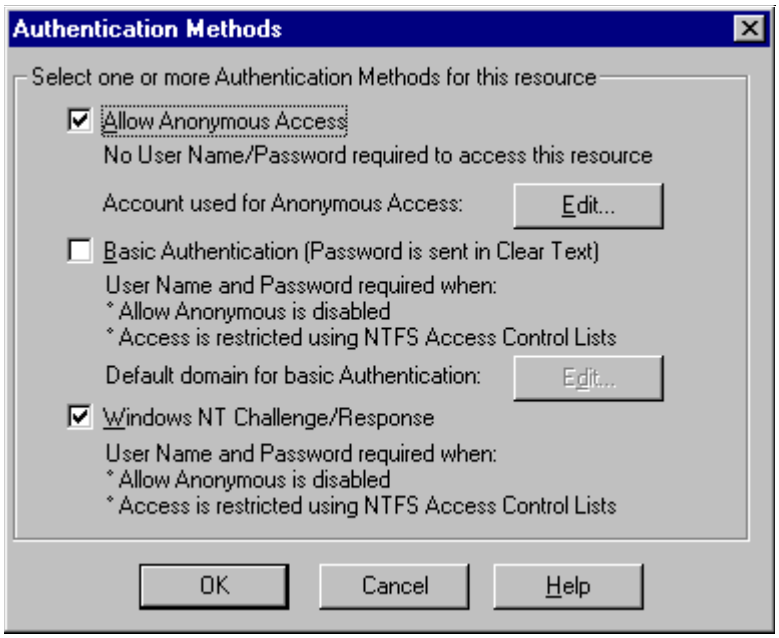

Fig. 6.4 Setting Authentication for loginverify.asp

3. Click **OK** on the other windows until you return to the Microsoft Management Console.
4. Close the Management Console.

Setting Permissions for the Presentation Folder

1. Use Windows NT Explorer to navigate to the ...getit\presentation folder.
2. Right-click on **presentation** and select **Properties**.
3. On the Security tab, click **Permissions**.

Note: If you do not see a Security tab, verify that Get.It! is installed on an NTFS partition.

4. Click **Add** to change the user groups that have permission to access the folder. Change the permission to a named authenticated group. For example, you could change permissions to all “Authenticated Users”.

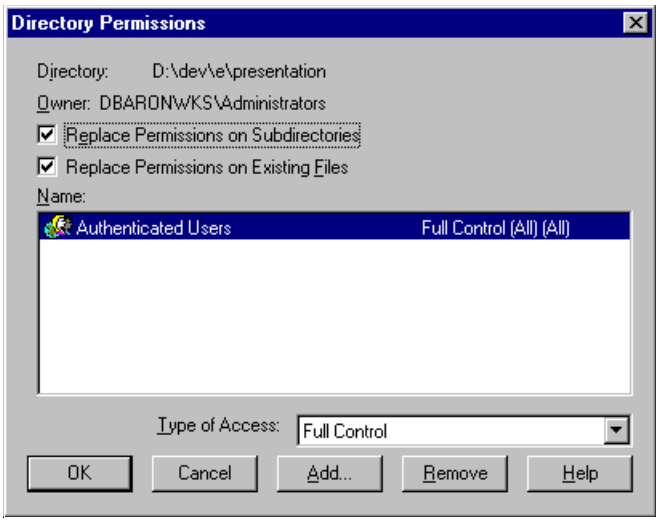

Fig. 6.5 Giving permissions to authenticated users

5. If the user group called “Everyone” has permissions, highlight the entry, and then click **Remove** so that only the group you selected in the previous step can access Get.It!.
6. Click **OK**. Close all remaining windows.

Testing the Settings

Log in to Get.It! to make sure the access permissions are set correctly. The Windows NT Challenge/Response settings are activated when you log in through a special login page named **login.asp**. Accessing Get.It! through the standard **login.htm** page results in the users needing to log on as usual.

1. Open a Web browser.
2. Enter the following URL: `http://webserver/getit/login.asp` in the browser address field (where `webserver` is the name of your Web server and `getit` is the name of the virtual directory created during installation).

3. Verify that access to Get.It! is what you expected based on the settings you chose for the **login.asp** and **loginverify.asp** files.

Setting the Default Login as login.asp

You can set the default login within Get.It! to use the Windows NT Challenge/ Response settings.

1. Open the **login.htm** file in the ...getit/presentation/ directory.
2. Locate the following:

```
function onPageLoad()
{
  top.location.replace( "login.jsp" );
}
</script>

<body onLoad="return onPageLoad();">
```

3. Change **login.jsp** to **login.asp**.
4. Save your changes.

Chapter 7
Adapters

Get.It! ships with several adapters you can use to connect to one or more databases. The following adapters are included with Get.It:

- AssetCenter adapter—provides a connection to Peregrine's AssetCenter.
- ServiceCenter adapter—provides a connection to Peregrine's ServiceCenter.
- JDBC adapter—allows you to establish a full database connection to databases other than AssetCenter or ServiceCenter. The JDBC adapter is used to connect the Get.It! demo weblication to the MS Access demo database.
- LDAP adapter—provides you with a centralized source for information about the people within an organization, eliminating the need to maintain user data in more than one location.
- E-mail adapter—allows you to connect to an external mail server.

Identifying the Back-end System Version

Document schemas can be modified to allow mapping to specific versions of the back-end systems. This API is available for use with the AssetCenter and ServiceCenter adapters. Each of these adapters use their DLL APIs to determine the version of the back-end system to which they are connected.

The extension is made possible by an optional “version” attribute that can be used as shown in the following example for AssetCenter:

```
<schema>
  <documents name="base">
     .... (base definition)
  </documents>

  <documents name="ac" version="3.02">
     .... (AC 3.02 definition)
  </documents>

  <documents name="ac" version="3.5">
     .... (AC 3.5 definition)
  </documents>
</schema>
```

The version of the back-end system is determined at run time using an Adapter API:

```
public double getVersion()
```

The following version rules apply when searching for a schema mapping:

- If a mapping has no version: automatically accepted as default.
- If mapping version > backend Version: mapping is rejected.
- If a mapping version <= backend version: mapping is accepted, but the DocManager keeps looking for the highest possible version.

In summary, if a schema and adapter have version information, the highest possible mapping version that is no higher than the current adapter version is used.

ServiceCenter Adapter

This section includes information about the **archway.ini** parameters that are specific to ServiceCenter, ServiceCenter event handling, and tips for troubleshooting the Get.It! connection to ServiceCenter.

Parameters

The **archway.ini** file contains parameters that are specific to ServiceCenter. The following table lists the parameters and gives a description of each.

Use the Settings page in the Get.It! Admin module to change these parameters if needed. The parameters are located on the Common tab in the ServiceCenter Adapter section.

Admin Module Field	Default Setting	Description
Admin Name	falcon	An administrator login must be defined in order to connect to ServiceCenter. The default is *falcon*, the sample administrator login supplied with ServiceCenter.
Admin Password		By default, no admin password is required.
Anonymous Name	falcon	With this login, requests sent to Archway are processed without going through the Get.It! user interface. Used by the scriptpoller function.
Anonymous Password		By default, no anonymous password is required.
Host	localhost	The host name for your ServiceCenter installation.
Port	12670	The port number for your ServiceCenter installation. The default port number for a full client is *12670*.
Log		The location of the **sclog** file.

ServiceCenter Event Handling

The ServiceCenter adapter provides a detailed *eventin* record that will give status when an event fails to execute or fails to create an eventout response.

The following is an example of an eventin record. The record was generated because an illegal approver tried to send an approval event:

```
<eventin>
   <evtype>approval</evtype>
   <evtime></evtime>
   <evsysseq>3991b1ff0036f001</evsysseq>
   <evusrseq>3991b1ff0036f001</evusrseq>
   <evsysopt></evsysopt>
   <evuser>Hartke</evuser>
   <evpswd></evpswd>
   <evsepchar>^</evsepchar>
   <evfields>^ocmq^Q1102^approve^^Hartke^^^</evfields>
   <evexpire>0</evexpire>
   <evstatus>error</evstatus>
   <evnumber></evnumber>
   <evlist></evlist>
   <evtimestamp>2000-08-09T19:40:05+00:00</evtimestamp>
   <evcount></evcount>
   <evnetnm></evnetnm>
   <evcode></evcode>
   <evmsg>
    <entry>Cannot execute application: es.approval</entry>
    <entry>Unrecoverable error in application:  es.approval on
    panel decide.exit</entry>
    <entry>You are not authorized to access Request Mgmt Quotes.
    </entry>
   </evmsg>
   <evid></evid>
   <sysmodcount>1</sysmodcount>
   <sysmoduser>N/A</sysmoduser>
   <sysmodtime>2000-08-09T19:40:05+00:00</sysmodtime>
</eventin>
```

The Archway script can use this information as necessary. For instance, it may display the *evmsg* error messages returned by the RAD application, or messages indicating that an event has failed because of customized Format Control rules.

Using the _event parameter

The "_event" parameter can be used in a script to specify the name of the event to be used in an SCDocManager operation.

If an _event parameter is not defined, the standard insert or update attributes of the document are used.

The following is an example of the use of the _event parameter:

```
var msgTicket = new Message( "Problem" );

...
msgTicket.set( "_event", "epmc" );
// Tell SCDocManager to use an epmc event for this update

archway.sendDocUpdate( "sc", msgTicket );
```

These lines will override the default insert and update attributes and instruct the SCDocManager to use the attributes defined in the script instead.

Troubleshooting the ServiceCenter Database Connection

If you are having trouble making a connection between Get.It! and the ServiceCenter database, verify the following:

1. Check the Control Panel page in the Admin module to confirm the database connectivity status. See "Verifying Adapter Connections" on page 2-24.
2. If “sc” is disconnected, verify that the ServiceCenter service is running (the ServiceCenter console has been started).
3. Verify that you have ServiceCenter *full* client connectivity by starting a client that is pointed to the port listed on the Settings page in the Admin module.

AssetCenter Adapter

This section includes information about the **archway.ini** parameters that are specific to AssetCenter, tips for troubleshooting the AssetCenter connection, and tells how to set AssetCenter feature links.

Information about AssetCenter adapter error codes is included in Chapter 8, Troubleshooting.

Get.It! INI Parameters

The following table lists the **archway.ini** AssetCenter parameters and gives a description of each. Use the Settings page in the Get.It! Admin module to change these parameters if needed.

Admin Module Field	Default Setting	Description
Admin Name	Admin	An administrator login must be defined in order to connect to AssetCenter. The default is *Admin*, the sample administrator login supplied with AssetCenter.
Admin Password	(none)	Password for the administrator login. By default, no admin password is required.
Anonymous Name	Admin	With this login, requests sent to Archway are processed without going through the Get.It! user interface. Used by the scriptpoller function.
Anonymous Password	(none)	Password for the anonymous login. By default, no anonymous password is required.
Database	(none)	The name of the AssetCenter database to which Get.It! will connect.

Troubleshooting the AssetCenter Database Connection

If you are having trouble making a connection between Get.It! and the AssetCenter database, verify the following:

1. Check the Control Panel page in the Get.It! Admin module to confirm the database connectivity status.

2. If “ac” is disconnected, verify that the “acdatabase” parameter in the **archway.ini** file is the same as the database name displayed when you log into AssetCenter. For example, in the system as shipped, the acdatabase parameter is set to ACDemo351ENG. Refer to your AssetCenter documentation for assistance with login procedures.
3. Verify that all AssetCenter settings match Get.It! settings. Log into the AssetCenter database. Make sure that the login account referenced in the Get.It! settings matches the login for AssetCenter. Also verify that Get.It! is using the correct user name and password for the connection. You can do this by selecting **File>Manage Connections** in AssetCenter.
4. Check the Get.It! ODBC connections to AssetCenter. Depending on the way you run JRun, it will look for either an ODBC User DSN or an ODBC System DSN.
   - If you start JRun as a service (the usual method), it references the System DSN for the ODBC connection to the AssetCenter database.
   - If you start JRun as an application, it references the User DSN to determine the ODBC connection.

AssetCenter Feature Links

AssetCenter feature links can be used in Get.It! schemas for read or write access in the same way as any other field or link from AssetCenter.

As an example of how this can be done, consider the following example of the fv_ShipToContact feature that implements a link to the employee table in the purchase order table. (See GRPurchaseOrder schema, Project.resources.Schemas.GRPurchaseOrder).

```
<attribute name="ShipToContactId" field="fv_ShipToContact" access="uiq"/>
<attribute name="RShipToContactId"field="fv_ShipToContact.lEmplDeptId"
    access="r"/>
<attribute name="ShipToContactDesc" field="fv_ShipToContact" access="r"/>
```

- The first line is used to set (update or insert) and search against a specific contact ID. It is a numeric field.
- The second line is used to read the purchase order's contact ID. It is also a numeric field.
- The third line returns the PO's contact description. This is a text field.

Compare this to how ShipToContact would be set up as a regular link in AssetCenter:

```
<attribute name="ShipToContactId"    field="lShipToCntcId"/>
<attribute name="ShipToContactDesc"  field="ShipToContact"
     access="r"/>
```

There is only one attribute that gives read/write/search access to the contact ID (numeric). It maps to the lShipToCntcId field in the amPOrder table. This field is the foreign key representing the link in the amPOrder table. In the implementation using a link feature, there is no access to a foreign key; therefore two attributes must be used.

The other attribute ShiptToContactDesc, does not differ in its implementation, whether you are using a link or a link feature.

JDBC Adapter

The Java Database Connectivity (JDBC) adapter allows you to create a connection between Get.It! and a database other than AssetCenter or ServiceCenter.

You can set up as many JDBC adapters as you need. The only limitation is that each adapter must point to a different database target string.

Adding and configuring a JDBC adapter requires several steps:

- Creating the connection between Get.It! and the database.
- Validating the connection.
- Adding the database settings to the Settings form.
- Creating a module in the portal so that information from the database will be displayed in the desired format.

The following procedures outline these steps, using the example of establishing a connection to a database that stores data about employees.

Adding a JDBC Adapter

There are two steps to adding a JDBC Adapter to Get.It!:

- Verify the Data Source Name (DSN) for the database to which you want to connect.
- Update the **archway.ini** file with information about the database.

Verifying the System DSN

The first step in adding the JDBC adapter is to verify the DSN for your database. You will need to add this information to the **archway.ini** file.

If you are configuring the JDBC adapter to connect the demo database to AssetCenter, it is important that you set up the correct type of DSN. Get.It! uses the following ODBC mappings:

- When Get.It! is running within a JRun executable launched as a service, a **System DSN** definition of the mapping is used.
- When Get.It! is running within a JRun executable launched in application mode, a **User DSN** definition is used.

This section includes instructions for creating a System DSN, however, you can use the same process to create a User DSN.

If you have not yet created a system DSN for your database, use the following procedure to do so.

1. Open the Control Panel (**Start>Settings>Control Panel**) and double-click the ODBC Data Sources icon.
2. Select the System DSN tab, and then click **Add**.

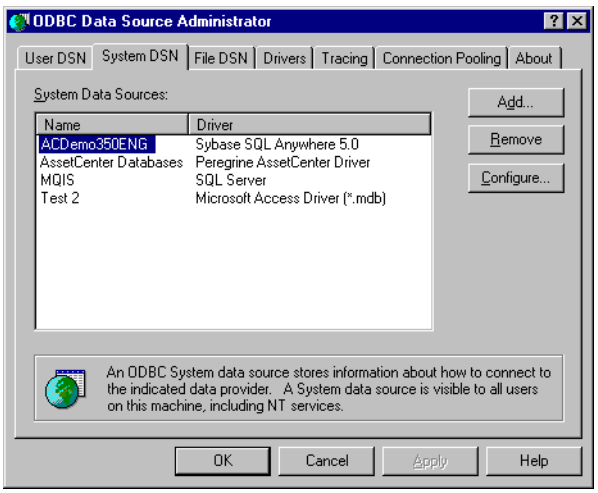

Fig. 7.1 ODBC Data Source Administrator

3. Select the driver for which you want to set up a data source. For this example, we are using the Microsoft Access Driver, because the example database was created using Microsoft Access.

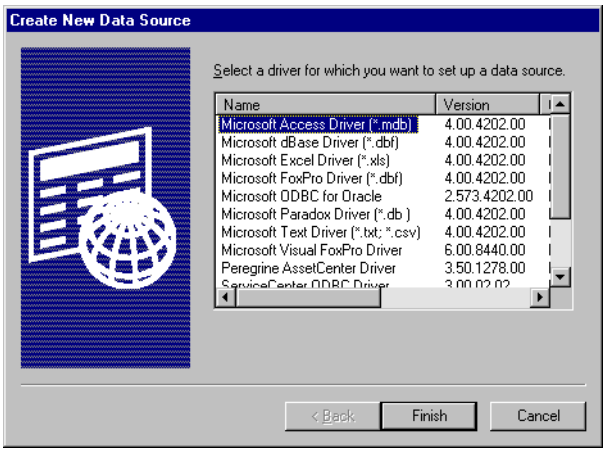

Fig. 7.2 List of available drivers

4. Click **Finish**.
5. In the dialog box displayed, give the data source a name. You can name it anything you want. For this example, we have used “Access_JDBC_test”.

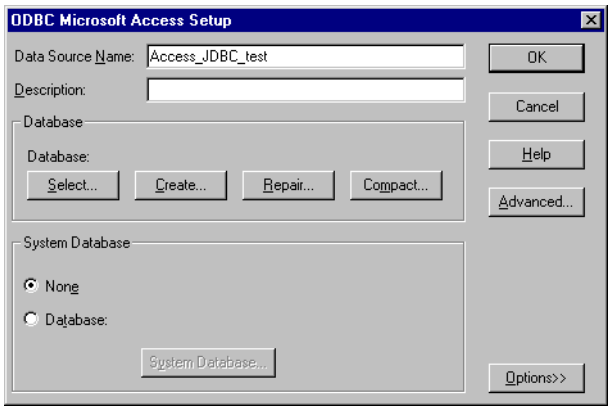

Fig. 7.3 ODBC setup—data source name

6. Click **Select**.

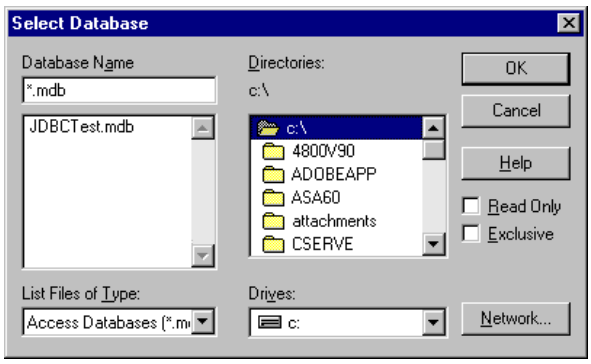

Fig. 7.4 Browse to the database location

7. In the dialog box displayed, select the directory location of the database to which you want to point the ODBC driver.

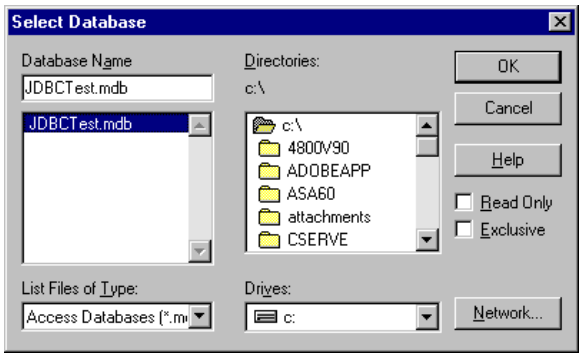

Fig. 7.5 Select the database name

8. In the list of databases, select the name of your database, and then click **OK**. The new DSN is added to the list of data source names.

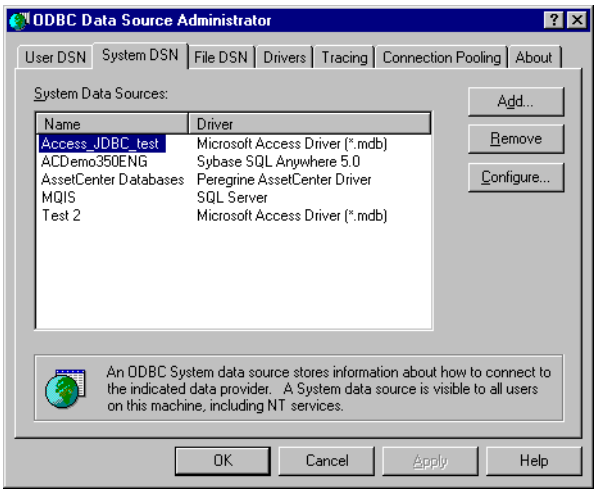

Fig. 7.6 New system DSN added

9. Click **OK**.

Updating the archway.ini File

Note: If you are configuring the JDBC adapter to be used with the Get.It! demo weblication, it is not necessary to complete the following procedure. The settings for the database are configured during the installation. The only parameter you need to set is `xxdatabase=<odbc name>`
If you make any changes to other settings, do not change the 2-letter database identifier as instructed in the procedure; it must remain “xx” for the demo weblication.

1. Open the **archway.ini** file in the `C:\getit\bin\` directory.
2. Update the adapters line with the JDBC Adapter, so the line would look something like:

   ```
   adapters=sc=SCAdapter;portalDB=SCAdapter;xx=JDBCAdapter
   ```

   where `xx` is a two character designation of the database to which you are linking.
3. Add the following lines to the **archway.ini** file. Replace `xx` with the two character designation of the database to which you are linking. Replace `<odbc name>` with the system DSN for your database.

```
xxdatabase=<odbc name>
xxdatabaseurl=jdbc:odbc:
xxdatabasedriver=sun.jdbc.odbc.JdbcOdbcDriver
xxcasesensitive=false
xxanonymous=

xxanonymouspassword=
xxadmin=
xxadminpassword=
```

For this example, the modified lines would look like this:

```
madatabase=Access_JDBC_test
madatabaseurl=jdbc:odbc:
madatabasedriver=sun.jdbc.odbc.JdbcOdbcDriver
macasesensitive=false
maanonymous=
maanonymouspassword=
maadmin=jsmith
maadminpassword=test
```

The fields can be updated after you have changed the Administration module weblication to display them. The `xxadmin` and `xxadminpassword` are used to actually connect to the database and, therefore, must be a valid database user.

The `xxanonymous` and `xxanonymouspassword` as well as names used in the login screen are used for Get.It! user sessions.

The `xxdatabaseurl`, `xxdatabasedriver`, and `xxcasesensitive`, and anonymous login fields are optional. All others are required.

4. Save the file.
5. Stop and restart the JRun Default Server and the JRun Admin Server (**Start>Programs>JRun 3.1**).

Validating the JDBC Adapter Connection

Once you have completed the steps for adding an adapter, you can verify that the connection has been made.

1. Go to the database you want to query and find a table against which you can run a query. Make note of the table name.
2. Type the following query in the address field of your Web browser, and then press Enter:

```
http://<hostname>/servlet/archway?<adapter name>.query&_table=
<table name>
```

where <name of adapter> is the two-letter designation you gave your database in the **archway.ini** file.

If the query produces an error message stating that your database cannot be found, go to the Admin Settings screen and verify that your JDBC adapter is set. If you make any changes, restart the JRun Servers.

For our example, we used a table called “Employees” in the sample database, so our query would look like this:

```
http://localhost/servlet/archway?ma.query&_table=Employees
```

3. In the dialog box displayed, select to open the file from its current location, and then click **OK**.
4. Select a text editor to display the data, and then click **OK**.

   Figure 7.7 shows an example of data displayed using Notepad.

Note: How the information displays depends on how your Web browser is set up to view XML documents. The query results may display in the browser itself rather than requiring a separate program.

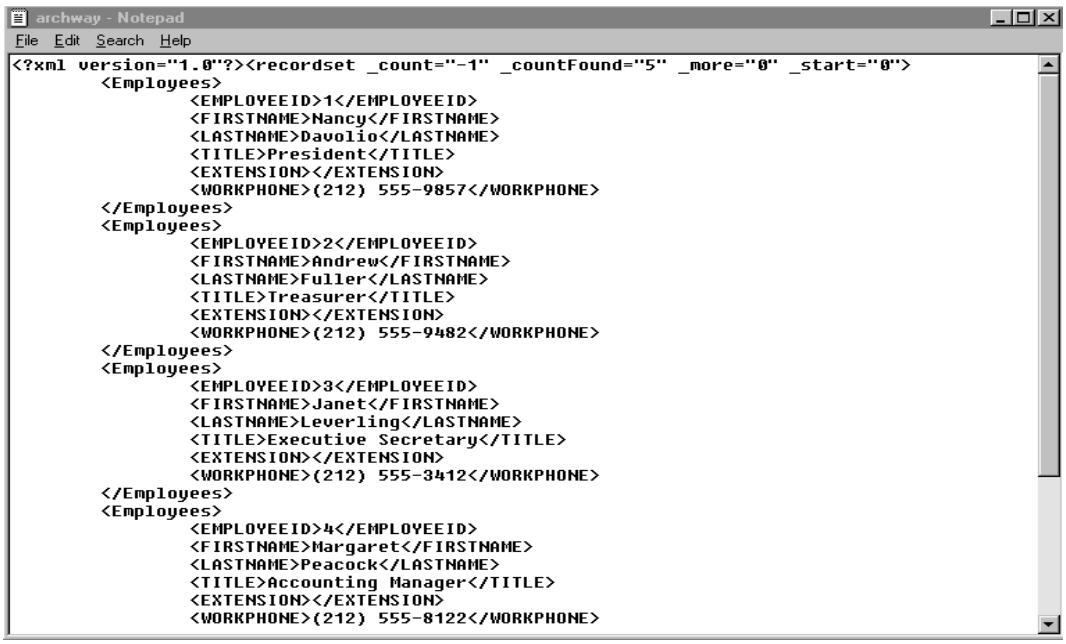

Fig. 7.7 Verification that the JDBC adapter is processing data

Adding the New Database Settings

This step is not required in order to set up the JDBC adapter. Perform this step only if you want to do JDBC adapter configuration using the Admin module.

The Settings form in the Get.It! Admin module contains the settings for the ServiceCenter and AssetCenter database connections. Although it is not necessary to add the JDBC adapter settings to the form in order to configure and use the adapter, it is a convenient way to make changes to the adapter settings.

You will need to update the **admin.xml** file if you want to add the JDBC adapter settings to this form. This can be done using Get.It! Studio, the tailoring utility for Get.It!. Refer to the *Get.It! Tailoring Guide* for instructions on using Get.It! Studio. You can use the adapter settings for the demo weblication JDBC adapter settings as a guide.

Note: If you have installed the demo weblication, these settings will already be added to the Admin module for this weblication. However, if you use a JDBC adapter to connect to a different database, you will need to create a new section on the Settings page for each JDBC adapter you configure.

Figure 7.8 shows the path to the **admin.xml** file in Get.It! Studio.

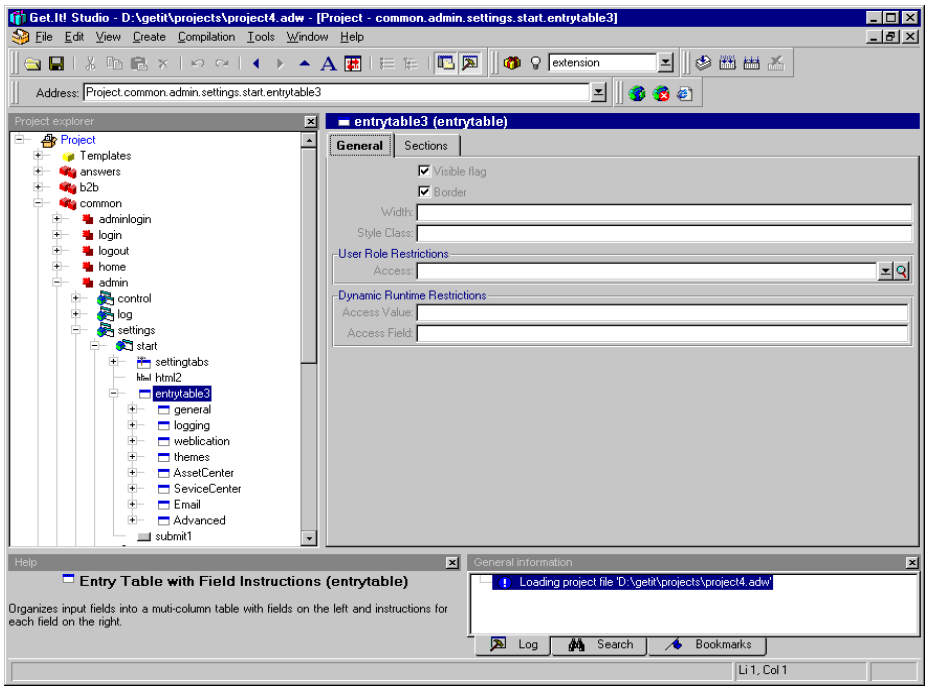

Fig. 7.8 Path to the admin.xml file in Get.It! Studio

Creating a New Weblication

In order to display formatted data from your database or to add new information from Get.It! to the database, you will need to create a new weblication using Get.It! Studio. The procedure for adding a new weblication includes the procedure for adding a label and image to the portal for access to the weblication. The *Get.It! Tailoring Guide* includes detailed instructions for creating a weblication.

Preventing Lost Connections

If you experience lost connections while using the JDBC adapter, you can use a polling script which will periodically check for database connectivity and reconnect when a connection is lost. The script is then run through **scriptpoller.ini** on a periodic basis.

The following script (**pingDB.js**) is an example. You will need to customize this script for your own database name and queries as noted. Replace the “test query” with one that you have used with your database and you know will succeed every time.

```
//---------------------------------------------------------------
// PingDB - attempts a well known query periodically, on error
// reconnects the JDBCAdapter to the database using the "_connect"
// event which will attempt a disconnect and then reconnect.
//
// Two functions are defined:
// start() - executes exactly once
// run()   - executes on the polling interval
//
//---------------------------------------------------------------
Archway = Packages.com.peregrine.archway;
Message = Archway.Message;

//---------------------------------------------------------------
// Start function .. can build parameters for run method
//---------------------------------------------------------------
function start( msg )
{
  var msgRet = new Message();
  msgRet.set("message","ok");
 return msgRet;
}

//---------------------------------------------------------------
// run function .. subsequent invocations
//---------------------------------------------------------------
```

```
function run( msg )
{
 var msgDB = new Message("test_query");
  var msgRes = null;
  var msgRet = new Message();

  msgDB.set("query", "select * from tab");
  msgDB.set("_count", "1");

  msgRes = archway.send( "xx", "query", msgDB );
  if (msgRes.get("message"))
  {
    msgDB = new Message("_connect");
    msgRes = archway.sendEvent( "xx", msgDB );
  }
  msgRet.set("message","ok");

  return msgRet;
}
```

Calling a Stored Procedure

Stored procedures can be called with input and output parameters. When the procedure is executed, the adapter first attempts to get any result sets that may have been returned, then attempts to retrieve the set of parameters with which the function was called. The resulting parameter list includes both input and output parameters and their values.

All parameters are treated as (and assumed to be declared as) Varchar or String type, with other data types to be supported in a future release.

Input and output parameters are specified with the “type” attribute; valid settings are “IN/OUT/INOUT”.

There are three ways to call a stored procedure:

- From the Web browser through CGI syntax in a URL.
- From a JScript file or post stream with an XML document containing _call and _sql tags.
- Using the Get.It! Document Manager.

The following is an example using a stored procedure called “insertEvent()” in an MSSQL7.0 database with two parameters; the first is of type “input” and the second, “output”. When called, the procedure verifies the input data and inserts a row returning the new row ID number.

Using a URL

```
http://localhost:8080/prgn/servlet/archway?ja.event
&_sql={call insertEvent('User1',?)}&name=User&value=User1&type=IN
&name=Id&type=OUT
```

Using a JScript

```
function processInsert( msg )
{
  var msgEvent = new Message("event");
  var strCall = "{CALL insertEvent('User1',?)}";

  msgEvent.set("_sql", strCall);
  msgArg = new Message("_arguments");

  elTag = msgArg.add("name", "UserName");
  elTag.setAttribute("type", "IN");
  elTag.setAttribute("value", "User1");

  elTag = msgArg.add("name", "Id");
  elTag.setAttribute("type", "OUT");

  msgEvent.add(msgArg);

  msg = archway.sendEvent("ja", msgEvent);

  return msg;
}
```

Using a Document Insert Or Update

This example uses the following schema definitions:

```
<?xml version="1.0"?>

<schema>

<!--================================================================
  Generic Schema Definitions
=================================================================-->

<documents  name="base">
  <document name="ProcEvent">
    <attribute name="UserName"       type="string"/>
    <attribute name="Id"             type="id"/>
  </document>
</documents>

<!--================================================================
  JDBCAdapter Schema Definition
=================================================================-->
```

```
<documents  name="ja">
  <document name="ProcEvent" table="None"  insert="insertEvent"
update="updateEvent">
    <attribute name="UserName" field="NAME"  insertEvent="IN"
updateEvent="IN"/>
    <attribute name="Id"       field="ID"    insertEvent="OUT"
updateEvent="IN"/>
  </document>
</documents>

</schema>
```

using the following JScript function:

```
function processInsert( msg )
{
  var msgRequest = new Message( "ProcEvent" );

  msgRequest.add( "UserName", "User1" );
  msgRequest = archway.sendDocInsert( "ja", msgRequest );

  return(msgRequest);
}
```

All methods above will produce the following output:

```
<?xml version="1.0" ?>
<_doc>
  <_arguments>
    <Id>915326</Id>
    <UserName>User1</UserName>
  </_arguments>
</_doc>
```

LDAP Adapter

Lightweight Directory Access Protocol (LDAP) directories provide a centralized source for information about the people within an organization. E-mail addresses, telephone numbers, fax numbers, user IDs, and passwords can be defined in the LDAP directory and referenced by various applications, eliminating the need to maintain user data in more than one location.

For example, the use of an LDAP directory and AssetCenter allows users without AssetCenter Employee Records to log on to Get.It! by providing minimal login information, such as login ID and password.

The information in this section assumes prior knowledge of the general setup and configuration of an LDAP adapter, and ability to write the associated JScript.

Connecting LDAP to Get.It!

There are three procedures that need to be completed in order to connect to Get.It! using LDAP:

- Update the **archway.ini** file to include the LDAP parameter.
- Create a JavaScript to query Get.It!.
- Reset the server.

This section provides information on updating the **archway.ini** file. Refer to your LDAP documentation for instructions on setting up an LDAP adapter. Sample scripts are not provided with Get.It! because of the variety of possible configurations. However, Chapter 4 of this guide includes instructions for modifying JScripts.

Updating the archway.ini File

To configure the LDAP adapter, update the **archway.ini** file with the following information:

1. Add the adapter name to the adapters line:

```
ac=ACAdapter;portalDB=ACAdapter;ldap=LDAPAdapter
```

2. Define the LDAP connection parameters:

```
ldapurl=xxx
ldaplogin=xxx
ldappassword=xxx
```

   Replace the “xxx” with your system and user-specific information.

E-mail Adapter

The E-mail adapter can be set up using Java or from an Archway ECMA script. The various calls that are available are the same either way. The E-mail adapter is based on the Sun Javax.mail classes for mail, especially:

- javax.mail.Session
- javax.mail.Message
- javax.mail.Transport
- javax.mail.search

and several other related classes. Detailed documentation on these is available from Sun's java.sun.com site.

The following script illustrates how the E-mail adapter can be used to connect to a mail server and how a given mailbox at that server can be processed:

```
// FESI definitions to point to the Archway mail adapter Java classes
  Archway  = Packages.com.peregrine.archway;
  MailAdapter = Archway.adapters.MailAdapter;

// Allocate a new MailAdapter object
  var ma0 = new MailAdapter();

// Optional: Turn on Javax.mail debug mode by calling
// "setDebug( true )"
// If this is done, copious javax.mail debug messages will be written
// to JRUN standard output log
  ma0.setDebug( true );

// Connect to a specified mailbox at a specified IMAP4-capable
// mail server, passing the hostname of the server, a userid,
// and a password
  ma0.setConnection( "exchange.mycorp.com", "joesmith",
  "opensesame" );

// If we could not connect for some reason, print MailAdapter last
// error in the log and return false
  if ( ma0.connect() != 0 )
   {
     var err = ma0.getLastError();

// This returns an Archway error message containing last javax.mail
// error info
     env.error( "Got error: " + err.getErrorMessage() +
     "trying to connect to mailbox" );
     return false;
    }
```

```
// Get a logical Archway lock on the mailbox we are going to process
  var strMailLock = "MailAdapter:" + "joesmith"
  env.getLock(  strMailLock  );

// Build the SQL query to query the mailbox
     var mailQuery = "SELECT * FROM INBOX ";
    if ( docType != null )
   {
      mailQuery += " WHERE subject = '" + docType + "' ";
   }
  if ( lastRecvDate != null && lastRecvDate != "" )
   {
      mailQuery += " AND receivedDate > " + lastRecvDate ;
   }

// Issue the query. Ask for 4 messages starting with zeroeth message.
// Specifying -1 for the count retrieves all messages
  var  m = ma0.doQuery( mailQuery, 0, 4 );

// Make sure it worked
  if ( m == null )
  {
       env.error(  ma0.getLastError() );
       env.releaseLock( strMailLock );
       return false;
  }

 env.debuglog( "Response from mailadapter was: " + m.getContent() );
  var i = 0;
  var docsProcessed = 0;
  var list = m.getList("message");
  for ( i=0; i < list.getLength(); i++ )
  {

// Get the mail message represented by this list entry
     var mm = ma0.getMailMessageByNumber( list.get( i,
     "messageNumber" ) );
     if ( mm == null )
     {
       env.log( "Failed to retrieve mail message by number - " +
       ma0.getLastError().getErrorMessage() );
       env.releaseLock( strMailLock );
       return false;
      }
     env.debuglog( "Mail message " + i + " is: " + mm.getContent
     ( false ) );

// Get a few elements out of the Mail message
     var msgID   = mm.get( "messageID" );
     var msgNo   = mm.get( "messageNumber" );
```

```
     var text    = mm.get( "text" );
     var subject = mm.get( "subject" );
     var dtReceived = new Date( mm.get( "receivedDate" ) );

//  Delete the Mail message
    ma0.deleteMailMessageByNumber( msgNo );
  }

// Commit inbox folder changes. This does a Java mail expunge
// operation
  ma0.commitInboxFolderChanges();

// Release lock on mailbox
  env.releaseLock( strMailLock );

// Close the mail box
  ma0.disconnect();
```

Verifying Adapter Connections

The status of a connection from Get.It! to a database can be verified from the Control Panel in the Admin module. In the example in Figure 7.9, “sc” is shown as disconnected because the ServiceCenter console was not started before logging into Get.It!.

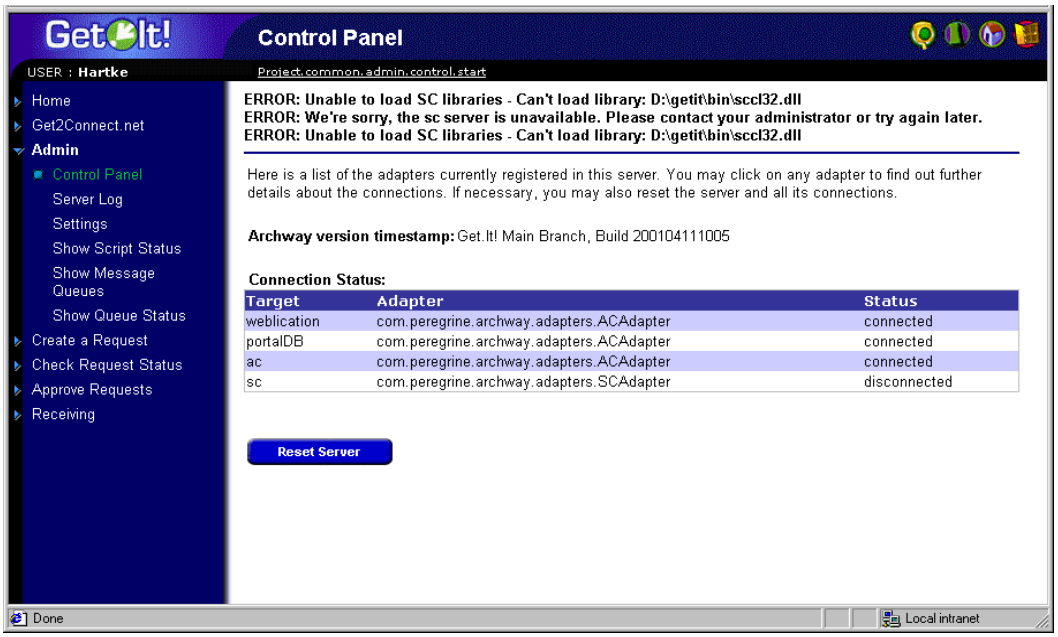

Fig. 7.9 Verifying adapter connections.

For more information on connectivity troubleshooting with each adapter, refer to the appropriate section on each Get.It! adapter provided in this chapter.

Chapter 8
Troubleshooting

This chapter includes information to help you troubleshoot the installation and operation of your Get.It! system.

Verifying Installation and Configuration

AssetCenter ODBC Error

If you are using the AssetCenter demo database shipped with Get.It! and AssetCenter with Sybase SQLAnywhere, you may get the following error message when trying to connect to AssetCenter with Get.It!:

```
ERROR: Unable to connect to AC: ODBC Error: [Sybase][ODBC
Driver]Unable to connect to database server:
unable to start database engine SQLSTate: 08001
ODBC: Unable to connect to database 'ACDemo351ENG' (UserId='itam')
unable to connect to this database engine.
```

This indicates that Get.It! is unable to connect to AssetCenter with both an AssetCenter client and a Get.It! client.

There are two possible solutions to this problem:

- Configure JRun to interact with the desktop

or

- Create a System DSN.

Configuring JRun to Interact with the Desktop

This procedure will enable JRun to connect Get.It! clients to AssetCenter using only a User DSN.

1. Go to **Start>Settings>Control Panel>Services**.

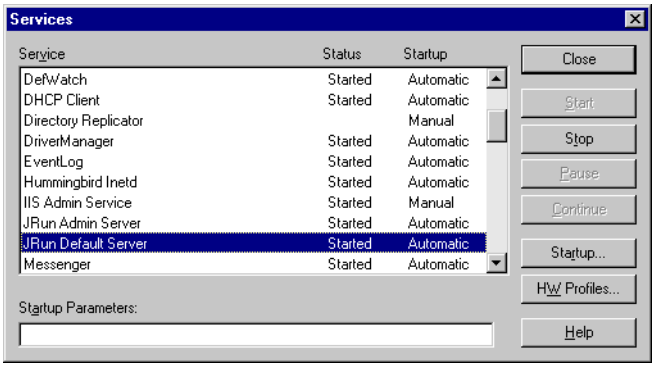

Fig. 8.1 Services

2. Find the entry for the JRun Default Server and double-click the entry.

   An edit dialog box is displayed.

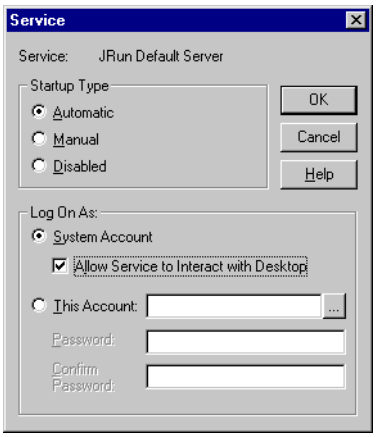

Fig. 8.2 Services edit dialog box

3. In the **Log On As** section, check **Allow Service To Interact With Desktop**. Click **OK**.
4. Restart the JRun Default Server service, and you should now be able to connect to AssetCenter.

Creating a System DSN

If you have configured JRun to interact with the desktop and you are still unable to connect to AssetCenter, you may need to create a System DSN.

Get.It! uses the following ODBC mappings:

- When Get.It! is running within a JRun executable launched as a service, a **System DSN** definition of the mapping is used.
- When Get.It! is running within a JRun executable launched in application mode, a **User DSN** definition is used.

If, when you install Get.It!, you point it to a database mapping created by AssetCenter's installation, both DSNs will already be in place.

However, if you choose to point AssetCenter to the demo database shipped with Get.It!, you must ensure that you create the correct DSN.

Refer to Chapter 7, "Adapters" for instructions for creating a DSN.

JRun Issues

Cannot Connect to JRun

If you try to access the Get.It! login page and get a “Cannot connect to JRun” message, there are several possible causes. However, there are a few steps you should take first which may solve the problem. These steps can be used whether you are attempting to connect after the initial installation or if you are encountering problems with an existing system.

The following sequence is suggested:

- Stop and restart the JRun Default Server and then the JRun Admin Server. (**Start>Programs>JRun 3.1>JRun Default Server / JRun Admin Server**). This may reestablish the connection with JRun.
- If cycling the JRun Servers does not enable connection, reinstall the Connector Wizard. The procedure for doing this is outlined below.
- Reboot your machine.

If you are still not able to connect to JRun, repeat the steps above. If this still does not solve the problem, reinstall IIS.

Reinstalling the Connector Wizard

If the JRun connector has been installed twice at different levels (for example, one globally and one or more at the virtual server level), this can cause a problem connecting to JRun. This can happen if you had a previous version of JRun 2.x or a beta/RC of JRun 3.1 installed at the global or local level, and you have run the Connector Wizard again since installing the JRun 3.1 final release and installed it at a different level.

If you want to keep previous local installations of the connector as they are, you do not need to remove all previous configurations of the connector. If the connectors are installed once at each local location, they will not conflict.

If you have installed the connector globally using JRun 3.1, then you must decide whether to keep the global configuration or the local ones. If you decide to install the connector globally, all Web sites will use the same connector.

To reinstall the Connector Wizard:

Task One: Stop the IIS Services

Stop the IIS Admin Service, which will also stop the World Wide Web Publishing Service and other IIS services (**Start>Settings>Control Panel>Services**).

Task Two: Delete Key Files

Once the IIS Admin Service is stopped, delete the following from your Web server's \scripts directory. The default location for the IIS scripts directory is C:\Inetpub\scripts:

- **jrun.dll**
- **jrun.ini**
- All associated **jrunNNNNNN.log** files (if desired)

Task Three: Remove the JRun ISAPI Filter

The next procedure removes the JRun ISAPI filter from your IIS configuration using the Microsoft Management Console (MMC) for IIS. This procedure removes all references to **jrun.dll** and the JRun Connector Filter.

To remove from the global level:

1. Right click the computer icon in the left hand side of the MMC, under the Internet Information Server folder, which is under the Console Root folder.
2. Select **Properties** from the context menu.

   A Properties dialog box is displayed.
3. In the Master Properties drop-down list, select WWW Service, and then click **Edit**.
4. In the WWW Service Master Properties dialog box, select the ISAPI Filters tab.

   If you have installed the JRun Connector Filter globally in the past, the filter will appear in the list of filters shown on this panel.
5. Select the JRun Connector Filter entry, and then click **Remove**.
6. Select the Home Directory tab.
7. In the Application Settings box, click the **Configuration** button.

8. In the Application Configuration dialog box, select the App Mappings tab. If there is a **.jsp** extension in the Application Mappings, select this entry and click **Remove**.

   If you want to map JSP to the **jrun.dll** for your default document, you can add this mapping back after installing the connector.

9. Click **OK** in all the open dialog boxes.
10. From the MMC Console menu, click **Save**.

To remove from the local, virtual server level:

1. Right click the top level folder for your Web site and select Properties. If you are using the Default Web Site that IIS set up for you, this is the Default Web Site folder.
2. In the Default Web Site Properties dialog box, select the ISAPI Filters tab.

   If you have installed the JRun Connector Filter globally in the past, the filter will appear in the list of filters shown on this panel.

3. Select the JRun Connector Filter entry, and then click **Remove**.
4. Select the Home Directory tab.
5. In the Application Settings box, click the **Configuration** button.
6. In the Application Configuration dialog box, select the App Mappings tab. If there is a **.jsp** extension in the Application Mappings, select this entry and click **Remove**.

   If you want to map JSP to the **jrun.dll** for your default document, you can add this mapping back after installing the connector.

7. Click **OK** in all the open dialog boxes.
8. From the MMC Console menu, click **Save**.

Task Four: Install The New Connector

1. Open the JRun 3.1 Java Management Console.
2. Log on and select **connector wizard**.
3. Follow the process outlined in the Get.It! installation procedures in Chapter 2 of this guide for using the connector wizard.

JRun Session Cookie

If the JRun session cookie is turned off, Get.It! will go into an infinite loop at login. To prevent this, ensure that the JRun session cookie is turned on and then stop and restart the JRun Default Server.

Turn the JRun session cookie on as follows:

1. In the JRun Management Console, go to **JRun Default Server>Web Applications>Default User Application>Web Application Session**.
2. Set the value for "Use Session Cookies" to *true*.
3. Stop and restart the JRun Default Server.

Log Files

There are a number of log files produced by JRun, ServiceCenter, AssetCenter, and Archway that provide information that can help you troubleshoot your Get.It! installation, including connectivity problems.

Although Get.It! does not operate at the Admin Server level, the admin logs are useful for locating JRun server errors. The Admin Server is used primarily to access the JRun Management Console. The default logs are especially useful for troubleshooting JRun/JSP problems.

Path to Log File	Description
\JRun\logs\admin-err.log	Error log for the JRun Admin Server.
\JRun\logs\admin-event.log	Event log for the JRun Admin Server.
\JRun\logs\admin-out.log	Output log for the JRun Admin Server.
\JRun\logs\default-err.log	Error log for the JRun Default Server.
\JRun\logs\default-event.log	Event log for the JRun Default Server.
archway.log	The location of this log file is specified in the **archway.ini** file. The information included in the log file is also determined by the INI file, but includes things like env.log, env.debuglog, and query information.
\Program Files\ServiceCenter\sc.log	Information regarding the ServiceCenter connection.

For more advanced debugging, use the JRun Management Console (**Start>Programs>JRun 3.1>JRun Default Settings>Log Settings**) to set the Debug level to “debug.” Be sure to cycle the JRun Admin and JRun Default servers after changing this setting.

AssetCenter Adapter Error Codes

The following table lists error codes that may be returned by the AssetCenter adapter. A description of the meaning of the code is included.

Code	Description
12001	Generic error.
12002	Function called with wrong parameter.
12003	Provided handle doesn’t match an existing object.
12004	No more data available in the result set.
12005	The database server signaled an internal error.
12006	Invalid value (incorrect type, out of range,...)
12007	Invalid record (mandatory field not given,...)
12008	No right to insert or update document.
12009	Function is obsolete or not yet implemented.
12010	Maximun number of connections reached (access denied).

Index

A

access rights
  AssetCenter 6-3
  ServiceCenter 6-3
adapters
  AssetCenter 7-6
  e-mail 7-21
  JDBC 7-8
  LDAP 7-20
  ServiceCenter 7-3
  using log files to troubleshoot 8-6
  verifying connections 7-24
Admin module 2-36
Admin Settings form
  adding JDBC adapter settings 7-15
  Content Type Encoding 2-52
  Currency 2-56
  Locale 2-53
administration
  archway.ini 5-6
  control panel 5-3
  control settings 5-6
  module 5-1
  resetting the server 5-3
  server log 5-4
application server, testing the installation 3-14
application servers, configuring 3-1
archway architecture
  building blocks 4-1
  clients 4-2
  diagram 4-2
  document manager 4-7
  executing queries against a system 4-7
  how it works 4-2
  query string 4-4
  requests 4-4
  XML 4-2
archway.ini 5-6
  parameters 7-4, 7-6
  updating for JDBC adapter 7-12
AssetCenter
  reconnecting to 5-3
AssetCenter adapter 7-6
  archway.ini parameters 7-6
  error codes 8-7
  setting feature links 7-7
  troubleshooting the DB connection 7-6
authentication
  overriding the login script 6-4
  user registration 6-2
  users 6-3

B

back-end system version 7-2

C

clients 4-2
connecting to JRun 8-3
Connector Wizard
  using 2-13
Connector Wizard, reinstalling 8-3
control panel 5-3
control settings 5-6
cookie, JRun session 8-6
country codes 2-53
currency codes 2-56

D

demo.adw file 2-24
document manager 4-7

E

e-mail adapter 7-21
encryption of passwords 6-3
error codes, AC Adapter 8-7
event handling in ServiceCenter 7-4

F

feature links in AssetCenter, setting 7-7

G

Get.It! Studio demo.adw 2-24

H

heap memory, setting 2-48, 3-3

I
ini file 5-6
installation
  Admin module configuration 2-36
  alternate application servers 3-1
  building demo.adw with Studio 2-24
  hardware required 2-3
  Java RTE 1.2.2 2-6
  reinstalling Get.It!
    Tomcat 3-3
    WebLogic 3-4
    WebSphere 3-12
  software requirements 2-2
  Solaris or Linux with JRun 2-28
  starting InstallShield 2-4
  using the Connector Wizard 2-13
  weblications 2-25
internationalization
  locale and currency codes 2-52
ISO currency codes 2-56
ISO locale codes 2-53

J
javaargs parameter 2-48
JDBC adapter
  adding 7-8
  adding to the Admin Settings form 7-15
  calling a stored procedure 7-17
  overview 7-8
  updating the archway.ini file 7-12
  validating the connection 7-13
  verifying the system DSN 7-9
JRun Connector Wizard
  reinstalling 8-3
  using 2-13
JRun session cookie 8-6
JRun, connecting to 8-3
JSPs, precompiling 3-7
JVMs, multiple 2-41

L
language codes 2-52
LDAP adapter
  connecting 7-20
  updating the archway.ini file 7-20
Linux Get.It! installation 2-28
load balancing 2-41
Locale codes 2-52
locale codes 2-53
log files 8-6
login script, overriding 6-4
login.asp 6-9

M
multiple JVMs or processors 2-41

O
ODBC errors with AssetCenter 8-1
overriding the login script 6-4

P
parameters
  setting for production 2-51
passwords, encryption 6-3

Q
query string 4-4

R
registering new users 6-2
reinstalling Get.It! on Tomcat 3-3
reinstalling Get.It! on WebLogic 3-4
reinstalling Get.It! on WebSphere 3-12

S
scriptpoller.ini files, updating 3-14
security
  alternate login authentication 6-4
  user authentication 6-3
  user registration 6-2
  Windows NT Challenge/Response 6-6
server
  log 5-4
  resetting 5-3
server processing considerations 2-41
ServiceCenter
  reconnecting to 5-3
  user access 6-3
ServiceCenter adapter 7-3
  _event parameter 7-4
  archway.ini parameters 7-4
  event handling 7-4
  troubleshooting the DB connection 7-5
settings 5-6
Solaris Get.It! installation 2-28
stored procedures, calling 7-17

T
Tomcat, configuring 3-1
troubleshooting
  AssetCenter ODBC errors 8-1
  cannot connect to JRun 8-3
    reinstalling JRun Connector Wizard 8-3
troubleshooting using log files 8-6

U
user registration 6-2
users
  access rights 6-3

V
version of back-end system 7-2

W
weblications, installing 2-25
WebLogic, configuring 3-1
WebSphere, configuring 3-1
WebSphere, precompiling JSPs 3-7
Windows NT Challenge/Response 6-1, 6-4
Windows NT Challenge/Response, configuring 6-6

X
XML 4-2

September 28, 2001