



# Creating a Runtime Environment



Sun Microsystems, Inc.  
4150 Network Circle  
Santa Clara, CA 95054  
U.S.A.

Part No: 820-3413-10  
June 2008

Copyright 2008 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun<sup>™</sup> Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

# Contents

---

|   |          |
|---|----------|
| <b>Creating a Runtime Environment .....</b>         | <b>5</b> |
| Building a Runtime Environment .....                | 6        |
| Using the Environment Editor .....                  | 6        |
| Using the Environment Context Menu .....            | 8        |
| Using Environment Component Context Menus .....     | 11       |
| Environment Components .....                        | 13       |
| Adding Environmental Constants .....                | 14       |
| Adding Logical Hosts and Domains .....              | 15       |
| Adding Application Servers .....                    | 17       |
| Adding Message Servers .....                        | 18       |
| Adding Scheduler External Systems .....             | 18       |
| Adding SOAP/HTTP Web Service External Systems ..... | 18       |
| Adding UDDI External Systems .....                  | 19       |



# Creating a Runtime Environment

---

Projects are run within *domains*, which contain the logical resources required by the Project at run time. Each domain contains a single instance of an application server and a single instance of a message server. The domains, in turn, are defined within runtime *Environments*, which represent the physical resources required to implement the Project. The Environment also contains information about external systems with which the Project interacts.

The following sections provide instructions on how to define a runtime Environment in which to run your Java CAPS Project. If you have any questions or problems, see the Java CAPS web site at <http://goldstar.stc.com/support>.

The basic procedure for setting up an Environment is described in “[Building a Runtime Environment](#)” on page 6.

Additional details about environmental components and relevant features of the user interface are provided in the following topics:

- “Using the Environment Context Menu” on page 8
- “Using Environment Component Context Menus” on page 11
- “Environment Components” on page 13
- “Adding Environmental Constants” on page 14
- “Adding Logical Hosts and Domains” on page 15
- “Adding Application Servers” on page 17
- “Adding Message Servers” on page 18
- “Adding Scheduler External Systems” on page 18
- “Adding SOAP/HTTP Web Service External Systems” on page 18
- “Adding UDDI External Systems” on page 19

# Building a Runtime Environment

You can build a runtime Environment by using the context menu system in the NetBeans Services window together with the Environment Editor. This section contains a brief outline of the procedure. Specific, in-context descriptions can be found in the various tutorials found in the Java CAPS documentation.

## Using the Environment Editor

The Environment Editor provides a canvas in which you create and customize a runtime Environment. Here you can see the various components (Logical Hosts, servers, and external systems) included in the selected Environment.

Clicking an Environment icon in the Services window invokes the Environment Editor, which provides a canvas in which you can create and customize an Environment.

Here you can see the various components (domains, servers, and external systems) included in the selected Environment. New Environments are added through the use of the CAPS Environments context menu. Components are added to the Environment by selecting options in the *Environment* and *Logical Host* context menus.

### ▼ To create and populate a runtime Environment

- 1 In the NetBeans Services window, right-click the CAPS Environments node to display its context menu.



FIGURE 1 Services Window: CAPS Environment Node

- 2 **Select New Environment, and assign an appropriate name.**
- 3 **Right-click the Environment to display its context menu, and create the components you need.**  
As a simple example, if you are creating an Environment to which to deploy the Project depicted in the following figure, you need the components listed below.

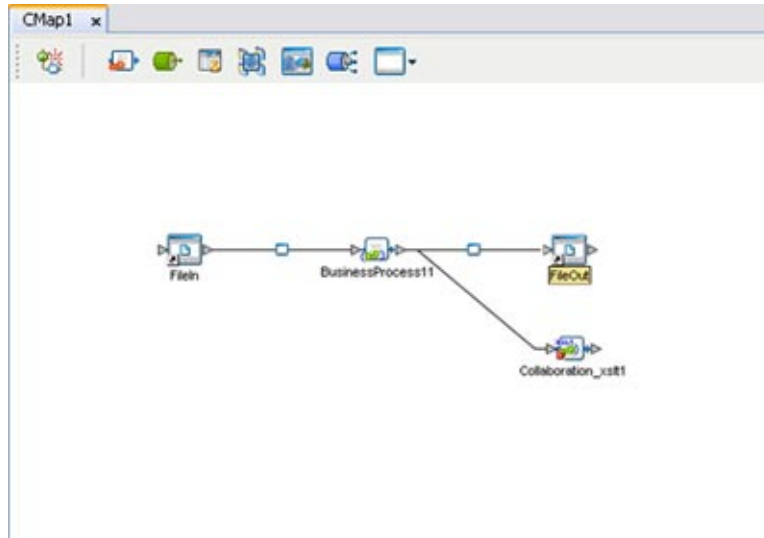


FIGURE 2 Example Project Connectivity Map

- a. A new Logical Host (see [“Adding Logical Hosts and Domains”](#) on page 15).
  - b. A new Application Server (see [“Adding Application Servers”](#) on page 17).
  - c. A new message server (see [“Adding Message Servers”](#) on page 18).
  - d. Two new File External Systems, one Inbound and one Outbound.
- 4 Name the Environmental components appropriately.
  - 5 Configure the components as described in the appropriate Configuration topic

## Using the Environment Context Menu

Right-clicking an *Environment* in the NetBeans Services window displays the context menu shown below, which lists the default set of components. Additional external systems may be displayed in your user interface, depending upon which adapters you have installed.



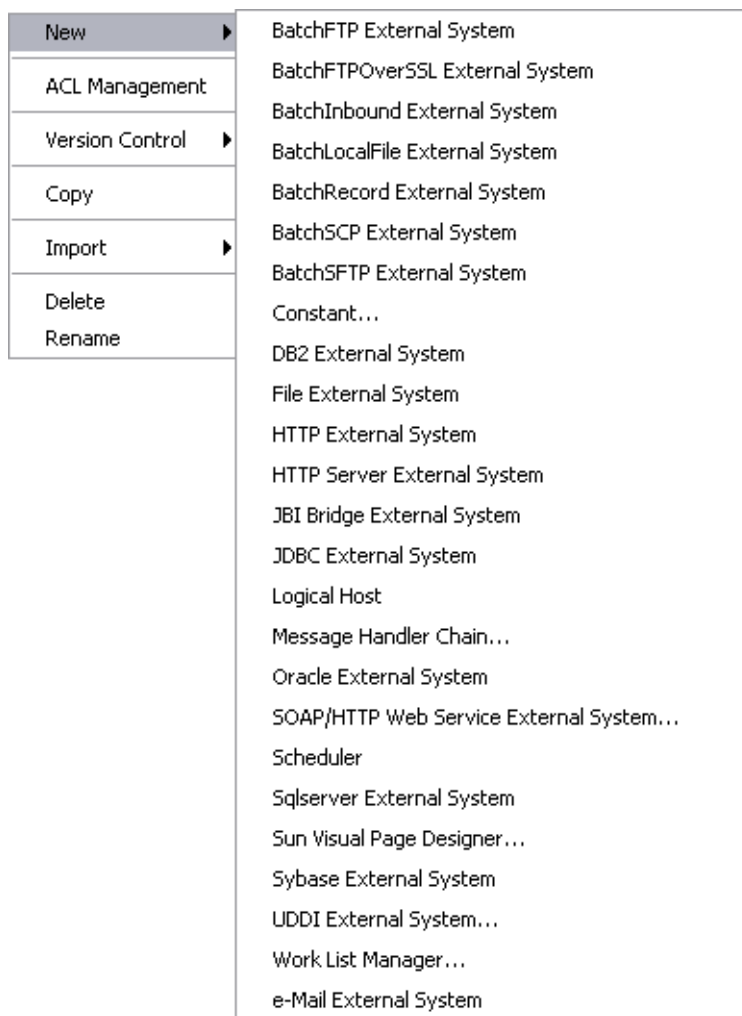


FIGURE 3 Environment Context Menu

External systems that are not listed in the following table are accessed through the corresponding adapter. You should consult the documentation for the specific adapter for information.

TABLE 1 Environment Context Menu Options

| Option          |                                       | Function  |
|-----------------|---------------------------------------|---|
| New             | Constant                              | Presents a dialog with which you can add a constant to the selected Environment. See <a href="#">“Adding Environmental Constants” on page 14.</a>   |
|                 | Logical Host                          | Adds a new Logical Host to the selected Environment. See <a href="#">“Adding Logical Hosts and Domains” on page 15.</a>   |
|                 | SOAP/HTTP Web Service External System | Adds a SOAP/HTTP Web Service External System to the selected Environment. See <a href="#">“Adding SOAP/HTTP Web Service External Systems” on page 18.</a>   |
|                 | Scheduler                             | Presents a dialog with which you can add a new scheduling component to the selected Environment. See <a href="#">“Adding Scheduler External Systems” on page 18.</a>  |
|                 | UDDI External System                  | Adds a UDDI External System to the selected Environment. See <a href="#">“Adding UDDI External Systems” on page 19.</a>   |
| ACL Management  |                                       | Presents a dialog with which an Administrator can assign read/write privileges to users for the selected Environment.   |
| Version Control | Version History                       | Presents a dialog with which you can track the version history for the selected Environment. The version history for an Environment that has been <i>cut</i> and pasted is preserved, since there can be only one instance of it. The version history for an Environment that has been <i>copied</i> and pasted is <i>not</i> preserved, since there can be multiple instances of it; the version number for each pasted instance is reset. |
|                 | Check In                              | Presents a dialog with which you can check in a new version of the selected Environment.  |
|                 | Check Out                             | Presents a dialog with which you can check out the current version of the selected Environment.   |
|                 | Undo Check Out                        | Presents a dialog with which you can reverse the last checkout of the selected Environment.   |
|                 | Make Latest                           | Presents a dialog with which you can check in the retrieved version of the selected Environment, making it the latest version.  |
|                 | Tag                                   | Presents a dialog with which you can specify a tag to attach to the selected Environment.   |

TABLE 1 Environment Context Menu Options (Continued)

| Option | Function  |
|--------|---|
| Copy   | Copies the selected Environment, including its components (“deep copy”), after which you can paste it to the same Repository branch multiple times. Once it has been pasted, you can rename it. All changes must be committed before you can copy the Environment. You can copy and paste an Environment even when another user has the Environment checked out.  |
| Import | Enables you to import a SOAP message handler archive.   |
| Delete | <p>Deletes the selected Environment, subject to the following conditions:</p> <ul style="list-style-type: none"> <li>■ You have <i>write</i> privileges for Environments (see ACL Management, above).</li> <li>■ The Environment is not checked out by anyone other than yourself.</li> </ul> <p>If these conditions are met, a dialog is presented in which you confirm that you want to delete the selected Environment. Clicking Yes then deletes the Environment.</p> |
| Rename | Activates the field, allowing you to rename the selected Environment.   |

---

**Note** – All Environment component names should contain only alphanumeric characters (letters and numbers), dashes, and underscores.

---

## Using Environment Component Context Menus

Right-clicking an *Environment component* in the NetBeans Services window displays its context menu. All components other than the Logical Host have a context menu such as that shown in the figure below.

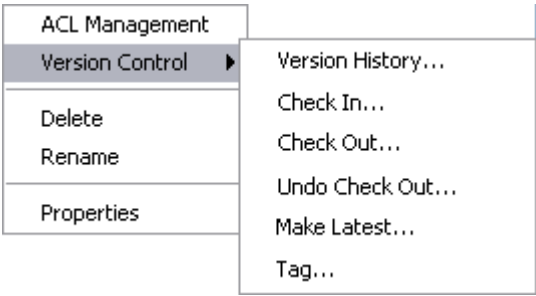


FIGURE 4 Environment Component Context Menu

TABLE 2 Project Component Context Menu Options

| Option          |                 | Function  |
|-----------------|-----------------|---|
| ACL Management  |                 | Presents the ACL Properties dialog, with which an Administrator can assign read/write privileges to users for the selected component.   |
| Version Control | Version History | Presents a dialog with which you can track the version history for the selected component. The version history for a component that has been <i>cut</i> and pasted is preserved, since there can be only one instance of it. The version history for a component that has been <i>copied</i> and pasted is <i>not</i> preserved, since there can be multiple instances of it; the version number for each pasted instance is reset. |
|                 | Check In        | Presents a dialog with which you can check in a new version of the selected component.  |
|                 | Check Out       | Presents a dialog with which you can check out the current version of the selected component.   |
|                 | Undo Check Out  | Presents a dialog with which you can reverse the last checkout of the selected component.   |
|                 | Make Latest     | Presents a dialog with which you can check in the retrieved version of the selected component, making it the latest version.  |
|                 | Tag             | Presents a dialog with which you can specify a tag to attach to the selected component.   |

TABLE 2 Project Component Context Menu Options (Continued)

| Option     | Function  |
|------------|---|
| Delete     | <p>Deletes the selected component, subject to the following conditions:</p> <ul style="list-style-type: none"> <li>■ You have <i>write</i> privileges for the component (see ACL Management, above).</li> <li>■ You have the component checked out to your workspace.</li> <li>■ The component is not checked out by anyone other than yourself.</li> </ul> <p>If these conditions are met, a dialog is presented in which you confirm that you want to delete the selected component. Clicking Yes then deletes the component.</p> |
| Rename     | Activates the field, allowing you to rename the selected component.   |
| Properties | Presents a dialog showing the configuration properties of the selected component, if such properties exist.   |

---

**Note** – All Environment component names should contain only alphanumeric characters (letters and numbers), dashes, and underscores.

---










## Environment Components

The CAPS Environment node in the NetBeans Services window deals with physical resources, including application and message servers, as shown in the following figure. The Environments it displays also contain information about external systems which may be involved with a Java CAPS configuration. The Services window is used in conjunction with the Environment Editor's canvas to create and configure the components of a runtime Environment.

Each component under an Environment node has an icon to identify the component type. Right-clicking on a component displays a context menu for that component, from which you can select various options. Only those menu options that are allowed for the component in its current state are enabled.

The icons described in the following table appear under each Environment node.

TABLE 3 Environment Component Icons

| Icon  | Function  |
|---|---|
|    | Represents a <i>Logical Host</i> or <i>Domain</i> , which contains the various logical components and files that are required at run time. See <a href="#">“Adding Logical Hosts and Domains” on page 15</a> .  |
|    | Represents a <i>Sun Java System Application Server</i> , which manages the Collaborations and other process interactions of a Project. The application server is deployed to a Logical Host (domain). See <a href="#">“Adding Application Servers” on page 17</a> . |
|    | Represents a <i>Sun Java System Message Queue</i> , which is the default message server for Java CAPS. It is used to store and forward system messages, and is deployed to a Logical Host. See <a href="#">“Adding Message Servers” on page 18</a> .                |
|    | Represents a <i>Sun JMS IQ Manager</i> , which is an alternate message server for Java CAPS, or a <i>Unified JMS Resource Adapter</i> . Both components are deployed to a Logical Host. See <a href="#">“Adding Message Servers” on page 18</a> .                   |
|    | Represents an <i>Environmental Constant</i> , which you can use to automate adapter and message destination configuration changes. See <a href="#">Defining Constants and Variables</a> for additional information.   |
|    | Represents a SOAP/HTTP Web Services External System, which acts as a container for a Web Services External Application. See <a href="#">Developing Java CAPS Projects</a> for additional information.   |
|    | Represents a UDDI External System, which acts as a container for a UDDI Registry.   |
|   | Represents an <i>External System</i> that is accessed through an adapter (a generic icon is shown). See the individual adapter documentation regarding these external systems.  |
|  | Represents a Work List Manager (WLM) External System, which acts as a container for a WLM.  |

## Adding Environmental Constants



Environmental constants are name-value pairs that are visible across the Environment. Selecting the New > Constant option from the Environment context menu presents the following dialog box, in which you define the constant.



FIGURE 5 Create Environmental Constant Dialog

See [Defining Constants and Variables](#) for additional information.

---

**Note** – When you create an Environmental constant, you assign a permanent value to it which cannot be overridden.

---

## Adding Logical Hosts and Domains



FIGURE 6 Logical Host Icon

You can create this component by selecting the **New > Logical Host** option from the *Environment* context menu in the NetBeans Services window.

A *domain* is an instance of a *Logical Host*; that is, the Logical Host used in Java CAPS acts as a directory for the domains, which are the actual runtime components. Each domain consists of two main components: an integration (or application) server and a message server.

All configuration of the individual domains is performed from either the Domain Manager or the command line.

**Note** – The Java CAPS runtime components, which include domains and any Projects or components deployed to them, provide their full functionality independent of the Repository.

Right-clicking a Logical Host in the NetBeans Services window displays the context menu shown below.

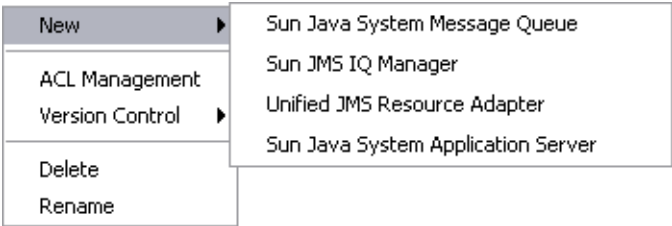


FIGURE 7 Logical Host Context Menu

TABLE 4 Logical Host Context Menu Options

| Option          |                                    | Function   |
|-----------------|------------------------------------|--|
| New             | Sun Java System Message Queue      | Adds a new Sun Java System Message Queue to the selected Logical Host. See <a href="#">“Adding Message Servers” on page 18</a> . |
|                 | Sun JMS IQ Manager                 | Adds a new Sun JMS IQ Manager to the selected Logical Host. See <a href="#">“Adding Message Servers” on page 18</a> .            |
|                 | Unified JMS Resource Adapter       | Adds a new Unified JMS Resource Adapter to the selected Logical Host.  |
|                 | Sun Java System Application Server | Adds a new Sun Java System Application Server to the selected Logical Host.  |
| ACL Management  |                                    | Presents a dialog with which an Administrator can assign read/write privileges to users for the selected Logical Host.           |
| Version Control |                                    | Accesses the version control features for the selected component.  |



TABLE 4 Logical Host Context Menu Options (Continued)

| Option | Function   |
|--------|--|
| Delete | <p>Deletes the selected Logical Host, subject to the following conditions:</p> <ul style="list-style-type: none"> <li>■ You have <i>write</i> privileges for the component (see ACL Management, above).</li> <li>■ You have the component checked out to your workspace.</li> <li>■ The component is not checked out by anyone other than yourself.</li> </ul> <p>If these conditions are met, a dialog is presented in which you confirm that you want to delete the selected component. Clicking Yes then deletes the component.</p> |
| Rename | <p>Activates the field, allowing you to rename the selected Logical Host.</p> <p>Note that all Environment component names should contain only alphanumeric characters (letters and numbers), dashes, and underscores.</p>   |

## Adding Application Servers



FIGURE 8 Application Server Icon

The Logical Host contains one or more instances of a Java EE-compatible application server. This server is the engine that runs Collaborations for processing business logic, and adapters for communicating with external applications. The application server provides services for security, transactions, business rules execution, and connectivity management. Java CAPS contains the Sun Java System Application Server and the Sun JMS IQ Manager.

You can add this component to the Environment by selecting the **New > [Application Server Type]** option from the *Logical Host* context menu in the NetBeans Services window, which presents a dialog box in which you enter the component name. Right-clicking the Application Server in the Service window displays its context menu. See [“Using Environment Component Context Menus” on page 11](#).

## Adding Message Servers



FIGURE 9 Message Server Icon

You add the message server component to the Environment by selecting the **New > [Message Server Type]** option from the *Logical Host* context menu in the NetBeans Services window, which presents a dialog box in which you enter the component name. Right-clicking the Message Server in the Service window displays its context menu. See [“Using Environment Component Context Menus”](#) on page 11.

## Adding Scheduler External Systems



FIGURE 10 Scheduler External System Icon

In an Environment, a *Scheduler* represents an external system to which you deploy a *Scheduler Adapter* created in a Project. Selecting the **New > Scheduler** option from the *Environment* context menu in the NetBeans Services window adds the Scheduler External System to the selected Environment. Right-clicking the Scheduler displays its context menu. See [“Using Environment Component Context Menus”](#) on page 11.

## Adding SOAP/HTTP Web Service External Systems



FIGURE 11 SOAP/HTTP Web Service External System Icon

A SOAP/HTTP Web Service External System represents a system containing a Web Services Application. You create this component by selecting the **New > SOAP/HTTP Web Service External System** option from the *Environment* context menu in the NetBeans Services window, which presents the following dialog box.



FIGURE 12 Create SOAP/HTTP WS External System Dialog

Entering a name, specifying whether it is a *Client* or *Server*, and clicking OK adds the SOAP/HTTP External System to the selected Environment. Right-clicking the SOAP/HTTP Web Service External System in the NetBeans Services window displays its context menu (see [“Using Environment Component Context Menus” on page 11](#)).

## Adding UDDI External Systems



FIGURE 13 UDDI External System Icon

A UDDI External System represents a UDDI Registry server in a runtime Environment. You create this component by selecting the **New > UDDI External System** option from the *Environment* context menu in the NetBeans Services window, which presents a dialog box in which you enter the component name.

---

**Note** – You can create only one UDDI External System per runtime Environment.

---

Entering a name and clicking OK adds the UDDI External System to the selected Environment. Right-clicking the UDDI External System in the NetBeans Services window displays its context menu. See [“Using Environment Component Context Menus” on page 11](#) for additional information.

