



Using SOAP Message Handlers



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

Part No: 820-4314-10
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 SunTM 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

Using SOAP Message Handlers	5
Importing a JAX-RPC Handler Archive	6
▼ To import an existing message handler archive	7
Creating a JAX-RPC Handler Chain	9
▼ To create a new chain of message handlers	9
Assigning Handler Chains to External Systems	11
▼ To assign the handler chain to a SOAP/HTTP Web Services External System	11

Using SOAP Message Handlers

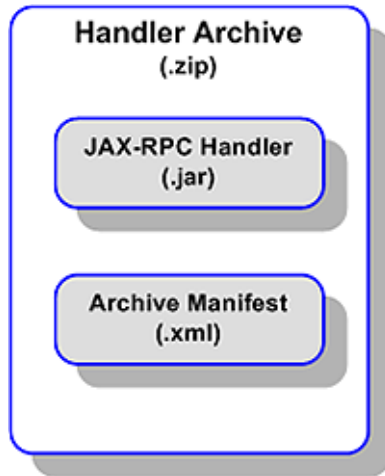
As defined in the JAX-RPC specification, message handlers provide a mechanism for adding, reading, and manipulating header blocks in SOAP messages that are sent and received by JAX-RPC clients and web service endpoints. By providing access to SOAP headers, you can import custom code to support SOAP standards that are not supported in basic versions of Java CAPS.

Although message handlers are Project-specific, they are added in Java CAPS Environments to allow easy reuse across multiple Projects. Multiple handlers can be installed or removed as needed, and they also may be chained, which allows staged execution of different types of processing modules.

Java CAPS version control is applied to all Environment sub-nodes and associated artifacts in the NetBeans IDE. You also may employ your own version control system for the handler code you import into Java CAPS.

JAX-RPC message handlers for use with Java CAPS are packaged in *handler archives*. A Handler Archive consists of a .zip file containing the following components:

- One or more JAX-RPC handlers
- An archive manifest file



JAX-RPC message handlers are incorporated into Java CAPS Projects by means of a three-stage process:

1. Importing a handler archive, which can contain one or more individual handlers.
2. Creating one or more handler chains, each of which can contain one or more individual handlers in a specific order.
3. Associating the handler chains with specific SOAP/HTTP Web Service External Services.

The following topics provide instructions on how to proceed through this process.

- “Importing a JAX-RPC Handler Archive” on page 6
- “Creating a JAX-RPC Handler Chain” on page 9
- “Assigning Handler Chains to External Systems” on page 11

If you have any questions or problems, see the Java CAPS web site at <http://goldstar.stc.com/support>.

Importing a JAX-RPC Handler Archive



FIGURE 1 Message Handler Archive Icon

To begin, you need to have a JAX-RPC handler archive available on your system.

▼ To import an existing message handler archive

- 1 From the Environment node's context menu, select Import > Message Handler Archive to invoke the Import Message Handler Archive wizard as shown in the following figure.



FIGURE 2 Import Message Handler Archive Wizard: File Name and Location Dialog

- 2 Type in or browse to the archive file you want to import, and click Next.
- 3 If the selected archive file imports correctly, you will see an Import Message Handler Archive wizard Summary page similar to that shown in the following figure.

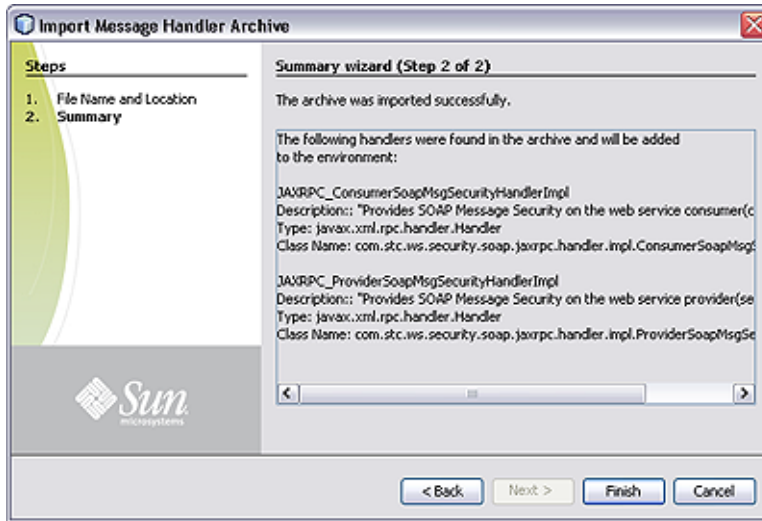


FIGURE 3 Import Message Handler Archive Wizard: Summary Window

The message handler archive now appears in the Services window as a child node of the Environment, and the message handler itself (in this case, a pair of them) as a child node of the archive, as shown in the following figure.



FIGURE 4 Message Handler Nodes

From the Message Handler Archive’s context menu, you can select Properties to display the Message Handler Properties.

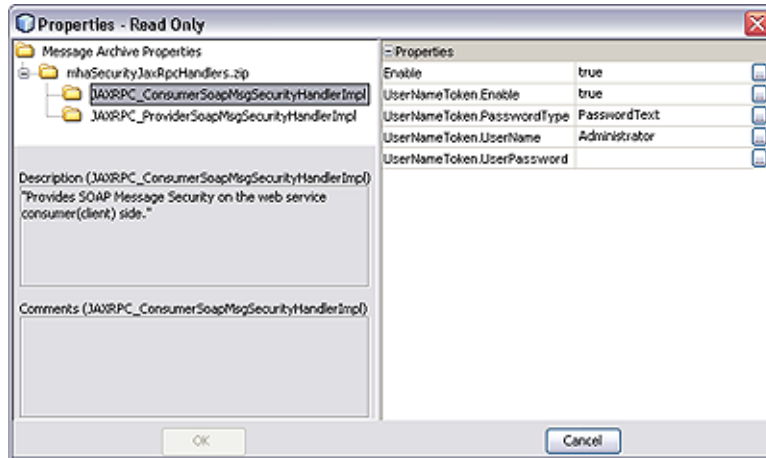


FIGURE 5 Example Message Handler Properties Dialog

- 4 If the archive file did not import properly, you will see an error message on the Import Message Handler Archive wizard Summary page. You will then need to determine the reason for the failure to import.

Creating a JAX-RPC Handler Chain



FIGURE 6 Message Handler Chain Icon

To begin, you need to have imported one or more JAX-RPC message handlers as described in [“Importing a JAX-RPC Handler Archive” on page 6](#).

▼ To create a new chain of message handlers

- 1 Select the **Services** tab to display the NetBeans Services window.
- 2 From the Environment node’s context menu, select **New > Message Handler Chain** to invoke the **Configure Handler Chain** wizard as shown in the following figure.

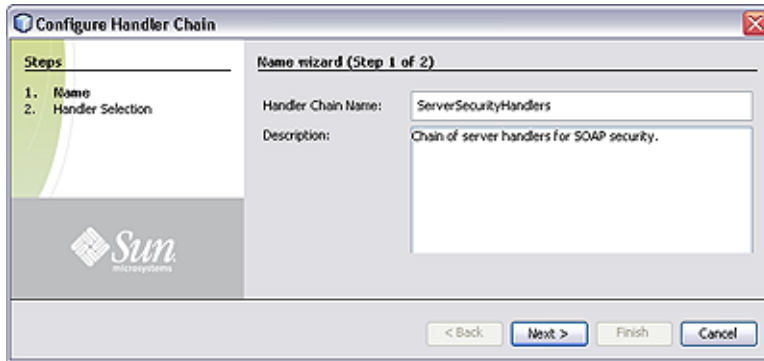


FIGURE 7 Configure Handler Chain Wizard: Name Dialog

- 3 Type in a name and a description for the handler chain you are creating.
- 4 Click Next to display the Handler Selection dialog, as shown in the following figure.

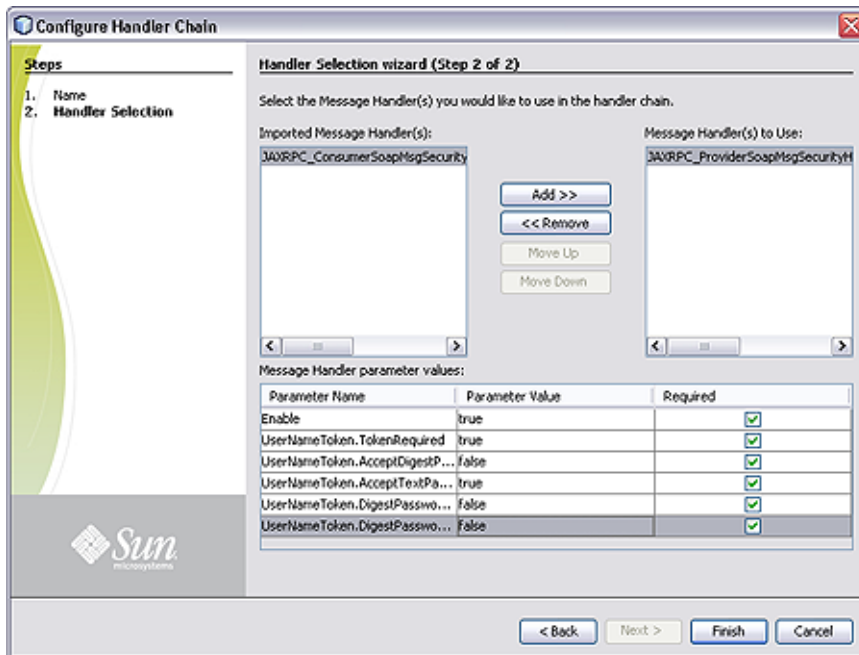


FIGURE 8 Configure Handler Chain Wizard: Handler Selection Dialog

- 5 A list of all imported message handlers appears in the left pane. Select the ones you want to include in the new chain, using the Add and Remove buttons to populate the right pane.

- 6 Use the **Move Up** and **Move Down** buttons to modify the order of the handlers in the chain. **Handlers are applied in the sequence you specify.**

Note – You can return to this wizard dialog to modify the settings at a later time by selecting **Edit** in the handler chain’s context menu.

The message handler chain now appears in the **Services** window as a child node of the **Environment**, as shown in the following figure.

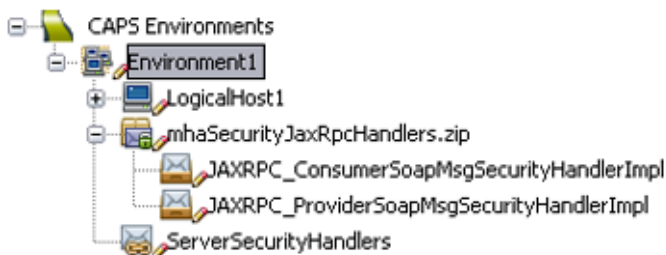


FIGURE 9 Message Handler Nodes

Note – You can define multiple handler chains containing different combinations of handlers, or different sequences of the same handlers.

Assigning Handler Chains to External Systems

To begin, you need to have created one or more handler chains as described in [“Creating a JAX-RPC Handler Chain”](#) on page 9

▼ To assign the handler chain to a SOAP/HTTP Web Services External System

- 1 Select the SOAP/HTTP Web Services External System to which you want to assign this handler.
- 2 Right-click the External System to display its context menu and select **Properties**, which displays its **Properties** dialog, as shown in the following figure.

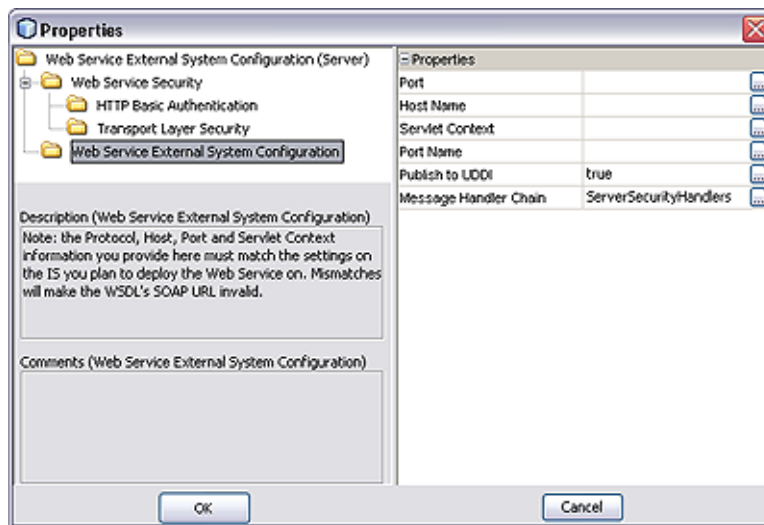


FIGURE 10 Web Services External System Properties Dialog

- 3 Click the button in the Message Handler Chain value field to open the drop-down dialog, and select the desired handler chain.
- 4 Click OK to assign the selected handler chain to this external system.