

Chapter 15

Configuring Applications to Communicate with an SAE with SDX Admin

This chapter discusses how to ensure that a Solaris platform is configured to support NIC proxies, how to test NIC applications in a Solaris platform, and how to monitor NIC on Solaris platforms. Topics Include:

- Reviewing and Updating the ORB Configuration for Applications That Include a NIC Proxy on Solaris on [page 207](#)
- Testing Applications by Using a NIC Proxy Stub on Solaris Platforms on [page 210](#)
- Monitoring NIC Proxies on Solaris Platforms on [page 213](#)

Reviewing and Updating the ORB Configuration for Applications That Include a NIC Proxy on Solaris

The JRE package (UMCjre) included in the SRC software distribution is preconfigured with JacORB.

JacORB meets the requirements for applications that include a NIC proxy. If you use a different JRE, you must ensure that it is configured with an ORB that supports value types with the Object Management Group's CORBA 2.6 standard.

If the default Java Virtual Machine (JVM) for the Web application server is UMCjre or another environment that complies with this standard, you do not need to configure the ORB. However, if this is not the case, you must configure the ORB to enable your application to communicate with the NIC.

Depending on the type of application, you can do one of the following:

- Configuring JacORB as the Default ORB on [page 208](#)
- Configuring One Web Application to Use JacORB on [page 209](#)
- Configuring a Web Application Server to Use JacORB on [page 209](#)

In this case, all Web applications, but not other Java applications, inside the Web application server will use this ORB.

For additional information about JacORB, see:

<http://www.jacorb.org/documentation.html>

For information about the Object Management Group's CORBA 2.6 standard:

<http://www.omg.org>

For information about how to set up the configurations for ORBs other than JacORB, see the documentation for that ORB.

For information about installing this package, see *SRC-PE Getting Started Guide, Chapter 33, Installing the SRC Software on a Solaris Platform*.

Configuring JacORB as the Default ORB

To configure JacORB as the default ORB for the JRE:

1. Access the folder in the SRC software distribution that contains the files you require for the version of JRE that you are using.

- For JRE 1.3, access the folder *SDK/lib-1.3*.

cd /cdrom/cdrom0/SDK/lib-1.3

- For JRE 1.4 or greater, access the folder *SDK/lib-1.4*.

cd /cdrom/cdrom0/SDK/lib-1.4

2. Copy the property files from the folder in the SRC software distribution to the folder *jre/lib* in your JRE installation.

cp jacorb.properties <jreInstallDirectory>/jre/lib/jacorb.properties
cp orb.properties <jreInstallDirectory>/jre/lib/orb.properties

3. Copy the appropriate JAR files from the folder in the SRC software distribution to the directory *jre/lib/ext* in your JRE installation.

- For JRE 1.3, copy the file *jacorb.jar*.

cp jacorb.jar <jreInstallDirectory>/jre/lib/ext/jacorb.jar

- For JRE 1.4 or greater, copy the file *jacorb.jar*.

cp jacorb.jar <jreInstallDirectory>/jre/lib/ext/jacorb.jar

Configuring One Web Application to Use JacORB

To configure a particular Web application that includes the NIC proxy to use JacORB:

1. Access the folder in the SRC software distribution that contains the files you require for the version of JRE that the Web application server is using.
 - For JRE 1.3, access the folder *SDK/lib-1.3*.

cd /cdrom/cdrom0/SDK/lib-1.3
 - For JRE 1.4, access the folder *SDK/lib-1.4*.

cd /cdrom/cdrom0/SDK/lib-1.4
2. Copy the appropriate files from the folder in the SRC software distribution to the folder *WEB-INF/lib* of the Web application.
 - For JRE 1.3, copy the files *jacorb.properties* and *jacorb.jar*.

cp jacob.properties <webAppDirectory>/WEB-INF/lib/jacob.properties
cp jacob.jar <webAppDirectory>/WEB-INF/lib/jacob.jar
 - For JRE 1.4, copy the files *jacorb.properties* and *jacorb.jar*.

cp jacob.properties <webAppDirectory>/WEB-INF/lib/jacob.properties
cp jacob.jar <webAppDirectory>/WEB-INF/lib/jacob.jar
3. Configure the NIC factory used by the Web application to use this ORB.

See *Chapter 16, Developing Applications That Use NIC*.

Configuring a Web Application Server to Use JacORB

To configure all Web applications, but not other Java applications, to use JacORB:

1. Access the folder in the SRC software distribution that contains the files you require for the version of JRE that the Web application server is using.
 - For JRE 1.3, access the folder *SDK/lib-1.3*.

cd /cdrom/cdrom0/SDK/lib-1.3
 - For JRE 1.4, access the folder *SDK/lib-1.4*.

cd /cdrom/cdrom0/SDK/lib-1.4
2. For JRE 1.3 and JRE 1.4, include the file *jacorb.jar* file in the classpath for the Web application server.
3. Include the file *jacorb.properties* for the appropriate JRE release in a directory specified in classpath, in the current directory, or in the home directory of the user who starts the Web application server.

4. Configure JacORB to be the ORB for the Web application server ORB. For information about this step, see the JacORB documentation at

<http://www.jacorb.org/documentation.html>

Testing Applications by Using a NIC Proxy Stub on Solaris Platforms

To test an application without NIC, you can configure a NIC proxy stub to take the place of the NIC. The NIC proxy stub comprises a set of explicit mappings of data keys and values in the namespace that contains the NIC proxy properties. When the SRC component passes a specified key to the NIC proxy stub, the NIC proxy stub returns the corresponding value.

For example, you can specify a subscriber's IP address that is associated with a particular SAE. When the SRC component passes this IP address to the NIC proxy stub, the NIC proxy stub returns the corresponding SAE.

Configuring a NIC Proxy Stub from SDX Admin

To use SDX Admin to configure a NIC proxy stub:

1. In the navigation pane, select the entry for the NIC proxy.
2. Add the following line to the NIC proxy properties.

**Gateway.nic.NicProxyClassName =
net.juniper.smgmt.gateway.gal.proxy.NicProxyStub**

For example, for Dynamic Service activator, located under *l = DynamicServiceActivation*, *l = WebApplication*, *ou = staticConfiguration*, *ou = Configuration*, *o = Management*, *o = umc*, you would add the lines similar to the following:

```
/nicProxies/ip/Gateway.nic.NicProxyClassName =  
net.juniper.smgmt.gateway.gal.proxy.NicProxyStub  
/nicProxies/ip/ANY_KEY = corbaloc::192.2.7.100:8801/SAE
```

When you use a NIC proxy stub, you must also configure test data for the stub to use.

See *Configuring the Test Data* on page 210.

Configuring the Test Data

To use a NIC proxy stub, you configure test data for the NIC proxy to use. You can specify that the test data indicate that any key return a specific SAE or that one or more keys map to particular values. If you specify an explicit SAE for a key, the NIC proxy stub returns the IOR for that SAE, rather than the value defined for the ANY_KEY property.

To configure test data, do one of the following:

- Configure a NIC proxy stub to use a corbaloc URL.

See *Configuring a NIC Proxy Stub to Use a corbaloc URL to Test Data* on page 211.

- Configure a NIC proxy stub to use a file URL.

See *Configuring a NIC Proxy Stub to Use a File URL to Test Data* on page 211.

- Configure a NIC proxy stub to use an IOR.

See *Configuring a NIC Proxy Stub to Use an IOR to Test Data* on page 212.

Configuring a NIC Proxy Stub to Use a corbaloc URL to Test Data

To configure a NIC proxy stub to use the corbaloc URL:

1. In the NIC proxy configuration, add a line in the format
corbaloc:: <host> : <port> /SAE

- <host> —Name or IP address of the SAE.

- <portNumber> —TCP/IP port number for the SAE. The default is 8801.

For example, corbaloc::127.0.0.1.145:8801/SAE.

2. In the NIC proxy configuration, add a line to return any key to a specific SAE or a key that the NIC proxy receives.

To return any key, add a line in the format
ANY_KEY = corbaloc:: <host> : <port> /SAE

For example, ANY_KEY = corbaloc::sae1:8801/SAE

To specify explicit mapping between keys and values, add lines in the following format to the NIC proxy configuration.

<mapping> = corbaloc:: <host> : <port> /SAE

For example, the following test data comprises two subscriber IP addresses associated with different SAEs. You define two explicit mappings:

```
192.0.2.10 = corbaloc::sae1:8801/SAE
192.0.2.11 = corbaloc::sae2:8801/SAE
```

Configuring a NIC Proxy Stub to Use a File URL to Test Data

To configure a NIC proxy stub to use the IOR file:

1. In the NIC proxy configuration, add a line in the format file:// <absolute path> to the IOR file.

For example, file:///opt/UMC/sae/var/run/sae.ior

2. In the NIC proxy configuration, add a line to return any key to a specific SAE or a key that the NIC proxy receives.

To return any key, add a line in the format ANY_KEY = file:// < absolute path to the IOR file > .

For example, ANY_KEY = file:///opt/UMC/sae/var/run/sae.ior

To specify explicit mapping between keys and values, add lines in the following format to the NIC proxy configuration.

< mapping > = file:// < absolute path to the IOR file >

For example, the following test data comprises two subscriber IP addresses associated with the same SAE. You define two explicit mappings:

```
192.0.2.0 = file:///opt/UMC/sae/var/run/sae.ior
192.0.2.1 = file:///opt/UMC/sae/var/run/sae.ior
```

Configuring a NIC Proxy Stub to Use an IOR to Test Data

To configure a NIC proxy stub to use a copy of the IOR:

1. Access the *sae.ior* file in the directory */opt/UMC/sae/var/run*.
2. Copy the complete IOR of the SAE from this file.
3. In the NIC proxy configuration, add a line to return any key to a specific SAE or a key that the NIC proxy receives.

To return any key, add a line in the format ANY_KEY = < SAE_IOR > .

■ < SAE_IOR > —IOR that you copied

For example:

```
ANY_KEY =
IOR:00000000000000003549444C3A736D67742E6A756E697065722E6E65742
F7361652F5365727669636541637469766174696F6E456E67696E653A312
E30000000000000002000000000000070000102000000000D31302E323
2372E312E323031000022610000001B5374616E64617264496D706C4E616
D652F736165504F412F53414500000000200000000000008000000004
A414300000000010000001C000000000010001000000010501000100010
1090000000105010001000000010000002C000000000000000100000001
0000001C0000000000001000100000001050100010001010900000001050
10001
```

To specify explicit mapping between keys and values, add lines in the following format to the NIC proxy configuration.

< key > = < value >

For example, the following test data comprises two subscriber IP addresses associated with different SAEs. You can define two explicit mappings:

```

192.0.2.0 =
IOR:00000000000000003549444C3A736D67742E6A756E697065722E6E65742
F7361652F5365727669636541637469766174696F6E456E67696E653A312
E300000000000000002000000000000070000102000000000D31302E323
2372E312E323031000022610000001B5374616E64617264496D706C4E616
D652F736165504F412F5341450000000002000000000000008000000004
A414300000000010000001C000000000010001000000010501000100010
1090000000105010001000000010000002C00000000000000100000001
0000001C000000000001000100000001050100010001010900000001050
10001
192.0.2.1 =
IOR:00000000000000002438444C3A736D67742E6A756E697065722E6E65742
F7361652F5365727669636541637469766174696F6E456E67696E653A312
E300000000000000002000000000000070000102000000000D31302E323
2372E312E323031000022610000001B5374616E64617264496D706C4E616
D652F736165504F412F5341450000000002000000000000008000000004
A414300000000010000001C000000000010001000000010501000100010
1090000000105010001000000010000002C00000000000000100000001
0000001C000000000001000100000001050100010001010900000001050
10001

```

Monitoring NIC Proxies on Solaris Platforms

You can use MBeans to monitor NIC proxies. MBeans are a feature of the Java Management Extension (JMX) software. If you want to monitor the MBeans for NIC proxies, your Web application server must include a JMX agent.

NIC proxies create one instance of an MBean called `NicProxyMgmt` to provide information about the role of the NIC proxy to the JMX agent. The way you view the MBeans depends on the particular Web application server and the interfaces that its JMX agent provides. Table 15 shows the information that this MBean provides.

You can reset the values of many `NicProxyMgmt` MBean properties to zero.

To reset the `NicProxyMgmt` MBean properties to zero:

- Execute the reset counters operation through the mechanism that the JMX agent for your Web application server provides.

Table 15 shows which counters the reset operation affects.

Table 15: Information That the `NicProxyMgmt` MBean Provides

Property	Description	Ability to Reset to Zero
<code>nicProxyName</code>	Name of the NIC proxy. Different NIC proxies may exist, providing different functionality.	No
<code>numKeysCachedLocally</code>	Number of key-value pairs that are cached in the NIC proxy (the bigger the cache, the less likely the NIC proxy will have to involve the distributed NIC components in lookups across the network).	No
<code>numLookups</code>	Number of times that the Web application containing this NIC proxy has requested the NIC proxy to look up a data key.	Yes
<code>numLookupErrors</code>	Number of lookups that have failed.	Yes

Table 15: Information That the NicProxyMgmt MBean Provides (continued)

Property	Description	Ability to Reset to Zero
numKeysNoMatch	Number of lookups in which the provided key does not map to any value.	Yes
numKeysOneMatch	Number of lookups in which the provided key maps to exactly one value.	Yes
numKeysMultiMatch	Number of lookups in which the provided key maps to more than one value.	Yes
lookupTimeAvg	For the 100 most recent (successful and unsuccessful) lookups, the average time (in milliseconds) of the lookup.	Yes
lookupTimeMin	For the 100 most recent (successful and unsuccessful) lookups, the minimum time (in milliseconds) of the lookup.	Yes
lookupTimeMax	For the 100 most recent (successful and unsuccessful) lookups, the maximum time (in milliseconds) of the lookup.	Yes