

## Chapter 13

# Configuring Applications to Communicate with an SAE

You configure NIC proxies for applications that use NIC. The NIC proxy configuration is part of the configuration for the application, and is not part of the configuration for a NIC host. This chapter discusses how to configure a NIC proxy for an SDX application.

The chapter contains the following sections:

- Before You Configure a NIC Proxy on page 245
- Overview of NIC Proxy Configuration on page 246
- Configuring NIC Proxies from SDX Configuration Editor on page 246
- Reviewing and Updating the ORB Configuration for Applications That Include a NIC Proxy on page 251
- Testing Applications by Using a NIC Proxy Stub on page 253

### Before You Configure a NIC Proxy

---

The values that you configure for a NIC proxy depend on the particular application; for example, you must specify the type of data used for the key and the type of data used for the value for each application. Before you configure a NIC proxy for an application, obtain the following information from the system manager who maintains the NIC configuration for NIC hosts:

- What is the name of the resolver that the application should use?
- What type of key will the application provide to the NIC host?
- What type of value is to be returned from the NIC host?
- Do NIC proxy cache values need to be changed? If so, what values should be supplied for the cache settings?
- Which groups should be listed for NIC host selection? These groups provide NIC replication.

Also, if you use a Java Runtime Environment other than the one included in the SDX Software distribution, review the configuration for the object request broker (ORB) to ensure that it meets the requirements for the NIC. See *Reviewing and Updating the ORB Configuration for Applications That Include a NIC Proxy* on page 251.

## Overview of NIC Proxy Configuration

---

For a number of SDX components, such as volume-tacking applications (VTAs) and the Dynamic Service Activator, you can configure the NIC proxy for the application from the Configuration Editor. For other applications, such as the sample residential portal, you configure the NIC proxy in a property file. If you configure a NIC proxy from a property file, the fields are the same as the fields that appear in SDX Configuration Editor. See *Configuring NIC Proxies from SDX Configuration Editor* on page 246.

When you develop and test SDX components that use a NIC, you can configure a NIC proxy stub to take the place of the NIC host. See *Testing Applications by Using a NIC Proxy Stub* on page 253.

For more information about NIC proxies, see *Chapter 12, Locating Subscriber Information*.

## Configuring NIC Proxies from SDX Configuration Editor

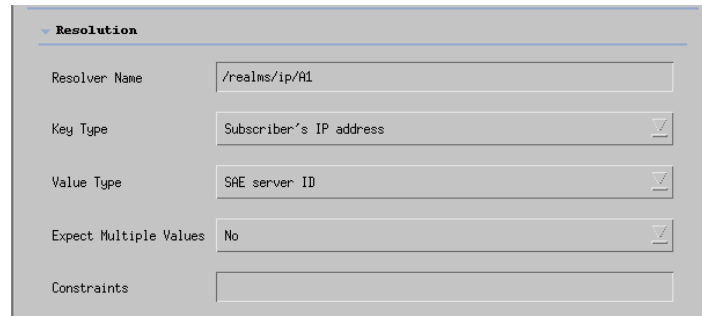
---

To configure a NIC proxy from SDX Configuration Editor:

1. In SDX Configuration Editor, access the configuration file for the application or SDX component for which you want to configure a NIC proxy.
2. Click on the tab that contains the NIC proxy configuration in the file.
3. Expand the entry called NIC proxy, and complete the fields under this entry (see *Configuring Resolution Information for a NIC Proxy* on page 247).
4. (Optional) Optimize the resolution performance of the NIC by configuring the NIC proxy's cache (see *Configuring the NIC Proxy Cache* on page 248).
5. (Optional but recommended) For NIC replication, configure the NIC proxy to communicate with groups of NIC hosts (see *Configuring the NIC Proxy for NIC Replication* on page 249).
6. Save the configuration.
7. Export the configuration file to the directory.

## Configuring Resolution Information for a NIC Proxy

In the NIC Proxy area of the pane, specify the information to be used by the NIC host to resolve a request. Use the following field descriptions to configure these properties.



Resolution	
Resolver Name	/realms/ip/RI
Key Type	Subscriber's IP address
Value Type	SAE server ID
Expect Multiple Values	No
Constraints	

### Resolver Name

- NIC resolver that this NIC proxy uses.
- Value—Path to NIC resolvers relative to the static configuration object in the directory
- Guidelines—This resolver must be the same as one that is configured on the NIC host.
- Default—No value
- Property—nic.server

### Key Type

- Type of data used the key provides for the NIC resolution.
- Value—Varies according to the application that is using the NIC proxy; choose from the menu.
- Guidelines—This key must be the same as the one for the specified resolver that is configured on the NIC host.
- Default—No value
- Property—nic.keytype

### Value Type

- Type of value to be returned in the resolution.
- Value—Varies according to the application that is using the NIC proxy; choose from the menu.
- Guidelines—This value must be the same as the one for the specified resolver that is configured on the NIC host.

- Default—No value
- Property—`nic.valuetype`

### **Expect Multiple Values**

- Whether or not the key can have multiple corresponding values.
- Value
  - Yes—Key may have multiple corresponding values.
  - No—Key may have only one value.
- Default—No
- Property—`nic.expectmultiple`

### **Constraints**

- Condition that must or may be satisfied before the next stage of the resolution process can proceed.
- Value—Comma-separated list of the data types of constraints specified for the NIC resolution on the NIC host.
- Guidelines—Provide a value only if the constraint will be provided by the application in the resolution request. Typically, you do not need to provide a value for this field.
- Default—No value
- Property—`nic.constraints`

## **Configuring the NIC Proxy Cache**

In the Cache area of the pane, you can modify cache properties for the NIC proxy to optimize the resolution performance for your network configuration and system resources. Typically, you can use the default values for the cache properties. Use the following field descriptions to configure these properties.

The screenshot shows a configuration pane for the NIC Proxy Cache. It has a title bar with a checkbox and the word 'Cache'. Below the title bar are three rows, each with a label and an input field. The first row is 'Cache Size' with a 'Disable' button. The second row is 'Cache Cleanup Interval' with an empty input field. The third row is 'Cache Entry Age' with a 'Disable' button.

### **Cache Size**

- Maximum size of the cache in which the NIC proxy retains data.
- Value—Integer in the range 0–4294967295
  - 0—Cache is disabled
  - Other values—Actual size of the cache

- Guidelines—You can change this value without restarting the NIC host. If you decrease the cache size or disable the cache while the NIC proxy is running, the NIC proxy removes entries in order of descending age until the cache size meets the new limit.
- Default—10000
- Property name—nic.maxCacheSize

### Cache Cleanup Interval

- Time interval at which the NIC proxy removes expired entries from its cache.
- Value—Number of seconds in the range 5–4294967295
- Default—15 seconds
- Property name—nic.cleanupInterval

### Cache Entry Age

- Maximum time that the NIC proxy can cache an entry. The NIC proxy compares this property with the life expectancy of each entry and uses the lower value to determine when to remove the entry.
- Value—Number of seconds in the range 0–4294967295
  - 0 or unspecified—Life expectancy of the data, which determines expiration of data
  - Other values—Actual time that the NIC proxy caches entries
- Default—0
- Property name—nic.maxCacheEntryAge

## Configuring the NIC Proxy for NIC Replication

In most cases, NIC hosts should be configured to use NIC replication. In the NICHost Selection area of the pane, specify the groups of NIC hosts to be contacted to resolve a request. Use the following field descriptions to configure these properties.

The screenshot shows a configuration window titled "NICHost Selection". It has a "Groups" text input field followed by a "Disable" button. Below that is a "Selection Criteria" text input field with a dropdown arrow. A section titled "Blacklisting" is expanded, showing three fields: "Try Next System on Error" with a checked checkbox, "Number of Retries Before Blacklisting" with a text input field, and "Blacklist Retry Interval" with a text input field.

**Groups**

- List of groups of NIC hosts that the NIC proxy can contact.
- Value—Comma-separated list of names of groups
- Default—No value
- Example—ontarioHost, vancouverHost
- Property name—nic.groups

**Selection Criteria**

- Selection algorithm that the NIC proxy uses to determine which NIC host to contact.
- Value
  - Round Robin—NIC proxy selects NIC hosts in a fixed, cyclic order. The NIC proxy always selects the next host in the list.
  - Random Pick—NIC proxy selects NIC hosts randomly from the list.
  - Priority List—NIC proxy selects NIC hosts according to their assigned priorities in the list. If the host with the highest priority in the list is not available, the NIC proxy tries the host with the next-highest priority, and so on.
- Guidelines—Use Round Robin or Random Pick to distribute resolution requests among NIC hosts. Use Priority List if you prefer to use a particular NIC host; for example, you may reduce operating costs by using a local NIC host.
- Default—Round Robin
- Property name—nic.repStrategy

**Try Next System on Error**

- Whether or not the NIC proxy should contact the next specified NIC host if a NIC host is determined to be unavailable.
- Value
  - Yes—NIC proxy should contact the next specified host.
  - No—NIC proxy should not contact the next specified host.
- Default—Yes
- Property name—nic.tryNextOnError

**Number of Retries Before Blacklisting**

- Number of times the NIC proxy tries to communicate with a NIC host before the NIC proxy stops communicating with the NIC host for a period of time.
- Value—Integer in the range 0–2147483647
- Default—3
- Property name—nic.numOfRetries

**Blacklist Retry Interval**

- Interval at which the NIC proxy attempts to connect to an unavailable NIC host.
- Value—Number of seconds in the range 0–2147483647
- Default—15
- Property name—`nic.retryInterval`

## **Reviewing and Updating the ORB Configuration for Applications That Include a NIC Proxy**

---

The Java Runtime Environment (JRE) package (UMCjre) included in the SDX software distribution is preconfigured with JacORB. For additional information, see:

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

For information about installing this package, see *SDX Software Basics Guide, Chapter 5, Installing the SDX-300 Software*.

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:

<http://www.omg.org>

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 252](#)
- [Configuring One Web Application to Use JacORB on page 252](#)
- [Configuring a Web Application Server to Use JacORB on page 253](#)

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

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

## Configuring JacORB as the Default ORB

To configure JacORB as the default ORB for the JRE:

1. Access the folder in the SDX 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 SDX 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 SDX 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 files *jacorb.jar*, and *smgttosf.jar*.
 

```
cp jacorb.jar <jreInstallDirectory>/jre/lib/ext/jacorb.jar  
cp smgttosf.jar <jreInstallDirectory>/jre/lib/ext/smgttosf.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 SDX 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 SDX 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 jacorb.properties <webAppDirectory>/WEB-INF/lib/jacorb.properties  
cp jacorb.jar <webAppDirectory>/WEB-INF/lib/jacorb.jar
```

- For JRE 1.4, copy the files *jacorb.properties*, *jacorb.jar*, and *smgttosf.jar*.

```
cp jacobr.properties <webAppDirectory>/WEB-INF/lib/jacobr.properties
cp jacobr.jar <webAppDirectory>/WEB-INF/lib/jacobr.jar
cp smgttosf.jar <webAppDirectory>/WEB-INF/lib/smgttosf.jar
```

3. Configure the NIC factory used by the Web application to use this ORB (see *Chapter 14, Developing Applications That Use a 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 SDX 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. Include the appropriate JAR files in the classpath for the Web application server.
  - For JRE 1.3, include the file *jacorb.jar*.
  - For JRE 1.4, include the files *jacorb.jar* and *smgttosf.jar*
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**

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 SDX 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 SDX component passes this IP address to the NIC proxy stub, the NIC proxy stub returns the corresponding SAE.

## Configuring a NIC Proxy Stub

To configure a NIC proxy stub from SDX Admin:

1. Start SDX Admin.
2. In the navigation pane, click on the entry for the NIC proxy.
3. Add the following line to the NIC proxy properties.

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

4. Configure the test data for the selected NIC proxy.

## Configuring the Test Data

You can specify that the test data indicate that any key return a specific SAE or that one or more keys map to specified 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.

You can configure test data by doing one of the following:

- [Configuring a NIC Proxy Stub to Use a corbaloc URL to Test Data on page 254](#)
- [Configuring a NIC Proxy Stub to Use a File URL to Test Data on page 255](#)
- [Configuring a NIC Proxy Stub to Use an IOR to Test Data on page 255](#)

### 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 form `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 form  
`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 form `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 form `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 form ANY\_KEY = <SAE\_IOR> .

- <SAE\_IOR> —IOR that you copied

For example:

```
ANY_KEY =
IOR:00000000000003549444C3A736D67742E6A756E697065722E6E65742F7361652F536572
7669636541637469766174696F6E456E67696E653A312E3000000000000002000000000000
0070000102000000000D31302E3232372E312E323031000022610000001B5374616E64617264
496D706C4E616D652F736165504F412F53414500000000200000000000008000000004A41
430000000001000001C00000000000100010000000105010001000101090000000105010001
000000010000002C000000000000001000000010000001C0000000000010001000000010501
0001000101090000000105010001
```

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:00000000000003549444C3A736D67742E6A756E697065722E6E65742F7361652F536572
7669636541637469766174696F6E456E67696E653A312E3000000000000002000000000000
0070000102000000000D31302E3232372E312E323031000022610000001B5374616E64617264
496D706C4E616D652F736165504F412F53414500000000200000000000008000000004A41
430000000001000001C00000000000100010000000105010001000101090000000105010001
000000010000002C000000000000001000000010000001C0000000000010001000000010501
0001000101090000000105010001
192.0.2.1 =
IOR:00000000000002438444C3A736D67742E6A756E697065722E6E65742F7361652F536572
7669636541637469766174696F6E456E67696E653A312E3000000000000002000000000000
0070000102000000000D31302E3232372E312E323031000022610000001B5374616E64617264
496D706C4E616D652F736165504F412F53414500000000200000000000008000000004A41
430000000001000001C00000000000100010000000105010001000101090000000105010001
000000010000002C000000000000001000000010000001C0000000000010001000000010501
0001000101090000000105010001
```