

## Chapter 15

# Configuring Applications to Communicate with an SAE with SDX Configuration Editor

This chapter discusses how to use SDX Configuration Editor to configure a NIC local host and NIC proxies for an SDX application. Topics Include:

The chapter contains the following sections:

- Configuring an Application to Use a Local NIC Host on page 243
- Configuring NIC Proxies from SDX Configuration Editor on page 245
- 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
- Monitoring NIC Proxies on page 258

### Configuring an Application to Use a Local NIC Host

---

If you want an application to use a local NIC host, you enable the local NIC host and configure associated properties.

To use SDX Configuration Editor to configure a NIC host that runs within the same application:

1. In the navigation pane, select a configuration file for the NIC proxy you want to configure.
2. Depending on the configuration file, select the **NIC proxy** tab or the **NIC proxy** section on another tab.

- Expand the **Local NIC Host Configuration** section.

The screenshot shows a configuration window titled "Local NicHost Configuration". It contains three main settings:

- Run NIC Host locally:** A dropdown menu currently showing "Yes".
- Local NIC Host Name:** A text input field containing the value "DemoHost".
- NIC static configuration IN:** A text input field followed by an "Enable" button.

- In the Local NIC Host Configuration section, edit or accept the default values for the fields.

See *Local NIC Host Configuration Fields* on page 244.

- Select **File > Save**.
- Right-click the configuration file, and select **System Configuration > Export to LDAP Directory**.

For more information about local NIC hosts, see *Chapter 9, Locating Subscriber Information with the NIC*.

### Local NIC Host Configuration Fields

In SDX Configuration Editor, you can modify the following fields in the NIC Host Configuration section of a NIC Proxy pane or section of a pane in a configuration file.

#### Run NIC Host Locally

- Specifies whether or not the NIC host is to run in the same application as NIC proxies.
- Value
  - Yes—NIC host will run in the same applications as a specified NIC proxy, or all NIC proxies.
  - No—NIC host will not run in the same application as NIC proxies.
- Default—No
- Property name—nic.localhost

#### Local NIC Host Name

- Name of the NIC host that is to run within the same application as the NIC proxy.
- Value— < Local NIC host name >
- Default—Name of the local NIC host; typically, DemoHost
- Property name—nic.hostname

**NIC Static Configuration DN**

- DN of the location in which the NIC configuration is stored.
- Value—DN
- Example—*l = OnePop, l = NIC, ou = staticConfiguration, ou = Configuration, o = Management, o = umc*
- Property name—`nic.staticConfigDN`

**Configuring NIC Proxies from SDX Configuration Editor**

Before you configure NIC proxies, see *Chapter 13, Configuring Applications to Communicate with an SAE*.

Tasks to configure a NIC proxy from SDX Configuration Editor are:

- Configuring Resolution Information for a NIC Proxy on page 245.
- For applications that use client/server mode (not NIC local host mode):
  - (Optional) Configuring the NIC Proxy Cache on page 247.
  - (Optional but recommended) Configuring the NIC Proxy for NIC Replication on page 249.

**Configuring Resolution Information for a NIC Proxy**

To use SDX Configuration Editor to configure a NIC resolution:

1. In the navigation pane, select a configuration file for the NIC proxy you want to configure.
2. Depending on the configuration file, select the **NIC proxy** tab or the **NIC proxy** section on another tab.
3. Expand the **Resolution** section.

The screenshot shows the 'Resolution' section of the SDX Configuration Editor. The section is expanded, revealing several fields:

- Resolver Name:** /realms/ip/R1
- Key Type:** Subscriber's IP address
- Value Type:** SAE server ID
- Expect Multiple Values:** No
- Constraints:** (empty field)
- Use Local NIC Host:** (checkbox, currently unchecked)

4. In the Resolution section, edit or accept the default values for the fields.

See *NIC Proxy Resolution Fields* on page 246.

5. Select **File > Save**.
6. Right-click the configuration file, and select **System Configuration > Export to LDAP Directory**.

### **NIC Proxy Resolution Fields**

In SDX Configuration Editor, you can modify the following fields in the Resolution section of a NIC Proxy pane or section of a pane in a configuration file.

#### **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 that 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.
- Default—No value
- Property—nic.valuetype

#### **Expect Multiple Values**

- Specifies whether or not the key can have multiple corresponding values.
- Value
  - Yes—Key can have multiple corresponding values.
  - No—Key can 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`

**Use Local NIC Host**

- Specifies whether or not the NIC proxy uses a NIC local host if one is configured and running within the application.
- Value
  - Yes—NIC hosts runs within the same application as the NIC proxies for the application.
  - No—NIC host can run on a machine other than the one on which the NIC proxies are located.
- Default—Yes
- Property—`nic.localhost`

**Configuring the NIC Proxy Cache**

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.

To use SDX Configuration Editor to configure NIC proxy cache properties:

1. In the navigation pane, select a configuration file for the NIC proxy you want to configure.
2. Depending on the configuration file, select the **NIC proxy** tab or the **NIC proxy** section on another tab.
3. Expand the **Cache** section.

The screenshot shows a configuration window with a tab labeled 'Cache'. Inside the tab, there are three rows of configuration fields:

- Cache Size:** A text input field with a 'Disable' button to its right.
- Cache Cleanup Interval:** A text input field.
- Cache Entry Age:** A text input field with a 'Disable' button to its right.

4. In the Cache section, edit or accept the default values for the fields.

See *NIC Proxy Cache Fields* on page 248.

5. Select **File > Save**.
6. Right-click the configuration file, and select **System Configuration > Export to LDAP Directory**.

### **NIC Proxy Cache Fields**

In SDX Configuration Editor, you can modify the following fields in the Cache section of a NIC Proxy pane or section of a pane in a configuration file.

#### **Cache Size**

- Maximum size of the cache in which the NIC proxy retains data.
- Value—Integer in the range 0–2147483647
  - 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 5–2147483
- 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 NIC Host Selection section of the pane, specify the groups of NIC hosts to be contacted to resolve a request.

To use SDX Configuration Editor to configure NIC replication for a NIC proxy:

1. In the navigation pane, select a configuration file for the NIC proxy you want to configure.
2. Depending on the configuration file, select the **NIC proxy** tab or the **NIC proxy** section on another tab.
3. Expand the **NIC Host Selection** section and the **Blacklisting** section.

The screenshot shows a configuration window with two main sections. The top section is titled 'NICHost Selection' and contains a 'Groups' text field with a 'Disable' button to its right, and a 'Selection Criteria' text field with a checkmark icon to its right. The bottom section is titled 'Blacklisting' and contains three text fields: 'Try Next System on Error' with a checkmark icon, 'Number of Retries Before Blacklisting', and 'Blacklist Retry Interval'.

4. In the NIC Host Selection section and the Blacklisting section, edit or accept the default values for the fields.

See *NIC Host Selection Fields* on page 249.

5. Select **File > Save**.
6. Right-click the configuration file, and select **System Configuration > Export to LDAP Directory**.

### NIC Host Selection Fields

In SDX Configuration Editor, you can modify the following fields in the NIC Host Selection section of a NIC Proxy pane or section of a pane in a configuration file.

#### 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**

- Specifies 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 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 251
- 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 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 28, 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 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* and *jacorb.jar*.

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

---

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 Configuration Editor

For applications that you can configure through SDX Configuration Editor, such as Dynamic Service Activator, you can configure a NIC proxy stub through the Editor.

To use SDX Configuration Editor to enable and configure a NIC proxy stub:

1. In the navigation pane, select a configuration file for the NIC proxy you want to configure.
2. Depending on the configuration file, select the **NIC proxy** tab or the **NIC Proxy** section on another tab.
3. Expand the **NIC Proxy** section, the section for an individual NIC proxy, and then the **Key-Value Pairs When Using NIC Proxy Stub** section.

The screenshot shows the SDX Configuration Editor interface. The 'NIC Proxy (ip)' section is selected and expanded. Under this section, there are three sub-sections: 'Resolution', 'Cache', and 'NICHost Selection'. Below these sub-sections is a 'NicProxyClass' field with a 'Disable' button. A checkbox labeled 'Key-value pairs when using NicProxyStub' is checked. Below the checkbox is a table with 'Property' and 'Value' columns. At the bottom are 'Add', 'Delete', and 'Refresh' buttons.

4. In the NIC Proxy Class field, select **Yes** for NIC Proxy Class, and specify the NIC proxy class for a NIC proxy stub, such as `net.juniper.smgmt.gateway.gal.proxy.NicProxyStub`.

See *NIC Proxy Stub Fields* on page 255.

5. Under Key-Value Pairs, enter value(s) that the proxy is to use in the format `key = value` and click **Add**.

See *NIC Proxy Stub Fields* on page 255.

6. Select **File > Save**.
7. Right-click the configuration file, and select **System Configuration > Export to LDAP Directory**.

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 256.

## NIC Proxy Stub Fields

In SDX Configuration Editor, you can modify the NIC Proxy Class field and the fields in the Key-Value Pairs When Using NIC Proxy Stub section in a NIC Proxy pane or section of a pane in a configuration file.

### NIC Proxy Class

- Specifies whether or not the application uses a specified NIC proxy class rather than an actual NIC proxy.
- Value
  - Enable—Enables a specified NIC proxy class for the application.
  - Disable—Disables a specified NIC proxy class for the application and enables an actual NIC proxy.
- Guidelines—Enable the specified NIC proxy class. If the NIC proxy class is a value such as `net.juniper.smgmt.gateway.gal.proxy.NicProxyStub`, the NIC proxy stub performs key-value resolutions. Other NIC configuration is not required. A NIC proxy stub is provided for testing purposes.
- Default—No value  
Property name—`NicProxyClass`

### Property

- Type of key for which the NIC proxy stub returns a value.
- Value— < key type >
- Example
  - IP address
  - Subscriber ID

### Value

- Value to be returned by the NIC proxy stub for a key specified in the Property field.
- Value— < key value >
- Example—SAE IOR

## 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 256.

## 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 256.

- 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 257.

- Configure a NIC proxy stub to use an IOR.

See *Configuring a NIC Proxy Stub to Use an IOR to Test Data* on page 257.

### 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
E300000000000000002000000000000070000102000000000D31302E323
2372E312E323031000022610000001B5374616E64617264496D706C4E616
D652F736165504F412F534145000000000200000000000008000000004
A414300000000010000001C000000000010001000000010501000100010
1090000000105010001000000010000002C000000000000000100000001
0000001C000000000001000100000001050100010001010900000001050
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
D652F736165504F412F534145000000000200000000000008000000004
A414300000000010000001C000000000010001000000010501000100010
1090000000105010001000000010000002C000000000000000100000001
0000001C000000000001000100000001050100010001010900000001050
10001
192.0.2.1 =
IOR:00000000000000002438444C3A736D67742E6A756E697065722E6E65742
F7361652F5365727669636541637469766174696F6E456E67696E653A312
E300000000000000002000000000000070000102000000000D31302E323
2372E312E323031000022610000001B5374616E64617264496D706C4E616
D652F736165504F412F534145000000000200000000000008000000004
A414300000000010000001C000000000010001000000010501000100010
1090000000105010001000000010000002C000000000000000100000001
0000001C000000000001000100000001050100010001010900000001050
10001
```

## Monitoring NIC Proxies

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

