Service Directory Maintenance

The SYSRPC utility provides functions used to maintain a service directory in order to connect the client's calling program to a subprogram on a server.

The service directory information is stored in the NATCLTGS subprogram and the XML-formatted SERVDIRX file (Natural text object) in the library that is defined with the profile parameter `RPCSDIR` (see the *Parameter Reference* documentation). If `RPCSDIR` is set, the service directory maintenance functions reference the library specified with `RPCSDIR`. If `RPCSDIR` is not set (this is the default), the library where you are logged on is referenced. In this case, log on to the library (or one of its steplibs) used by the client at runtime before you perform a service directory maintenance function.

The name of the library referenced for service directory maintenance is indicated in the root node of the service directory tree view (see *Tree Nodes*). If `RPCSDIR` is set, the root node contains **Central**, which indicates that the library displayed in the window is *not* the library where you are currently logged on, but the central library specified with `RPCSDIR`.

For further information on how to apply the service directory maintenance functions, refer to *Specifying RPC Server Addresses* described in *Operating a Natural RPC Environment* in the *Natural Remote Procedure Call (RPC)* documentation.

Attention:

If NATCLTGS is stored in the Natural system library SYSRPC, we strongly recommend that you move NATCLTGS to the application library (or one of its steplibs) used by the client.

This section covers the following topics:

- Service Directory Concept
- Tree Nodes
- Logon Option
- Transport Protocol

Service Directory Concept

The main items of a service directory are node, server, library and service (subprogram). The hierarchical structure of these items is displayed as a tree view in the **SYSRPC - Remote Procedure Call** window (see also *Service Directory Tree* in *Basic SYSRPC Functionality*). The highest level (root node) of the tree view is **Service Directory** and the lowest is service (subprogram).

The node and server names specified in the service directory are either physical names or logical names and logical services.

This section covers the following topics:

- Physical Nodes and Servers

- Location Transparency
- Example of a Service Directory

Physical Nodes and Servers

Physical node and server names denote the names of real nodes (valid TCP/IP or Entire Net-Work addresses) and servers.

In the *Example of a Service Directory* below, two servers are defined for one node. Both servers are connected to the same node: `ETB045`. The remote `CALLNAT` to subprogram `SUB1` is executed on server `NRPC001`, whereas subprograms `SUB2` and `SUB3` are executed on server `NRPC002`.

The server names specified here must be identical to the server names specified for the server with the profile parameter `SRVNAME` described in the *Parameter Reference* documentation. Analogously, the node name in the service directory must be identical to the node name specified for the server with the profile parameter `SRVNODE` described in the *Parameter Reference* documentation.

Location Transparency

Location transparency is a concept where physical node names can be replaced by logical node names, and a combination of physical node and server names can be replaced by logical services.

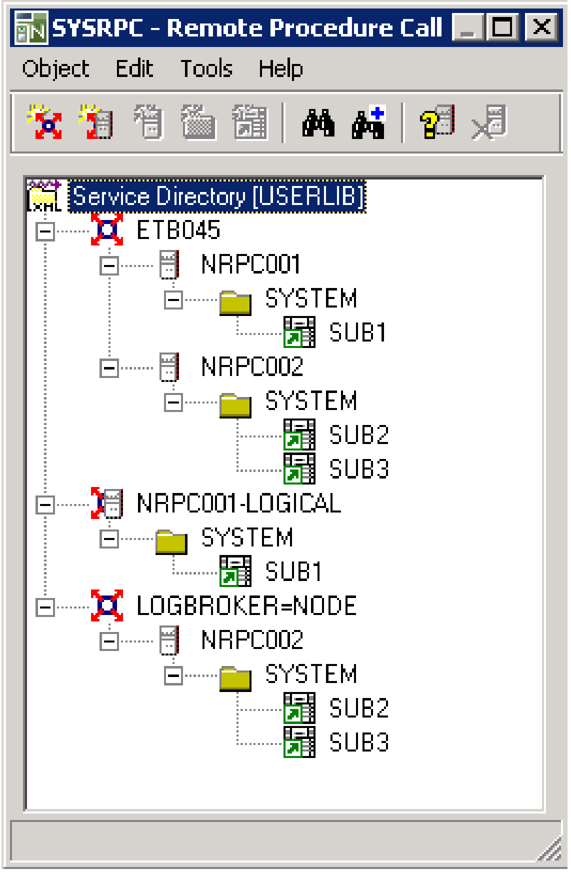
Logical node names and logical services are defined with EntireX and are assigned to physical node and server names at Natural runtime.

See also *Example of a Service Directory* and *Using Logical Services and Logical Node Names*.

Related Topics:

- *Using Location Transparency* in *Operating a Natural RPC Environment* in the *Natural Remote Procedure Call (RPC)* documentation.
- The relevant sections in the EntireX documentation.

Example of a Service Directory



In the example of a service directory above, the icon indicates that `NRPC001-LOGICAL` is a logical service. `LOGBROKER=NODE` indicates that `NODE` is the logical node name.

Tree Nodes

This section describes the tree nodes contained in the service directory tree view. Each tree node is identified by a different icon.

You can manipulate the tree nodes by using the functions provided by the menu bar, the toolbar and the context menu described in *Basic SYSRPC Functionality*.

Icon	Tree Node	Explanation
	Service Directory root	The service directory root node indicates the name of the library from which the service directory was read: **Service Directory [*library-name*]**. For example: If you invoked the SYSRPC utility from the library USERLIB, the root reads **Service Directory [USERLIB]**. If the profile parameter `RPCSDIR` is set, the root node reads **Central Service Directory [USERLIB]**, which indicates that the service directory is read from a central library specified with `RPCSDIR`. For explanations of other root node names that can occur, see *Root Node Names in Error Situations*.

Icon	Tree Node	Explanation
	Node	The name of the node to which the remote CALLNAT is sent. Maximum name length: Physical nodes: 32 characters Logical nodes: 192 characters Depending on the setting of the **Logon Option**, either of the following icons is displayed: Logon = No Logon = Yes See also **Logon Option** below.
	Server	The name of the server on which the CALLNAT is to be executed. Maximum name length: 32 characters Depending on the setting of the **Logon Option**, either of the following icons is displayed: Logon = No Logon = Yes See also *Logon Option* below.
	Logical Service	The name of a logical service. Maximum name length: 192 characters Depending on the setting of the **Logon Option**, either of the following icons is displayed: Logon = No Logon = Yes See also *Logon Option* below.
📁	Library	SYSTEM or the name of the library to which your client application is logged on during the execution of the remote CALLNAT.
	Service (Subprogram)	The name of the remote subprogram to be accessed from the client. Maximum number of entries: 500 subprograms.

The section below contains information on:

- Root Node Names in Error Situations
- Selection Criteria for Node and Server
- Using Logical Services and Node Names

Root Node Names in Error Situations

This section lists root node names that can occur if subprograms or text objects required by the service directory are missing, explains possible reasons and provides resolution advice.

Node Name	Reason	Resolution
Service Directory from NATCLTGS [*library-name*]	The text object SERVDIRX is missing. This is indicated by the icon .	From the **Object** menu, choose **Save As** or **Save**. SERVDIRX is generated into the library for which SYSRPC was invoked and the name of the root node changes to **Service Directory [*library-name*]**.
Example Service Directory	The subprogram NATCLTGS and the text object SERVDIRX are missing.	From the **Object** menu, choose **Save As** or **Save**. NATCLTGS and SERVDIRX are generated into the library for which SYSRPC was invoked and the name of the root node changes to **Service Directory [*library-name*]**.
Empty tree	NATCLTGS, SERVDIRX and the subprogram DEF-GS (delivered in the Natural system library SYSRPC) are missing. DEF-GS contains example data.	1. Create at least one new item for a node and a server or create at least one logical service. 2. Save the modifications. NATCLTGS and SERVDIRX are generated into the library for which SYSRPC was invoked and the name of the root node changes to **Service Directory [*library-name*]**.

Selection Criteria for Node and Server

At Natural runtime, the selection of a node and a server depends on the value of the service (subprogram) and library tree nodes. Comply with the following conditions:

Non-conversational `CALLNAT`

1. The library tree node must contain the name of the current application library or SYSTEM.
2. The subprogram referenced in the `CALLNAT` statement must be contained in the service (subprogram) tree node, which belongs to the library tree node in point (1).

Conversational `CALLNAT`

1. The library tree node must contain the name of the current application library or SYSTEM.
2. All subprograms specified in the `OPEN CONVERSATION` statement must be contained in a service (subprogram) tree node, which belongs to the library tree node in point (1).

The node and server used for a non-conversational or conversational `CALLNAT` are taken from the superior node and server tree nodes of the library tree node in point (1).

Using Logical Services and Node Names

You can define logical services and logical node names. For an example of logical services and logical node names, see *Example of a Service Directory*.

To define a logical service

- Select the **Service Directory** root node, open the **Edit** menu and choose **New Item** and **Logical Service (EntireX)**.

  Or:
  Select the **Service Directory** root node and choose **New Item** and **Logical Service (EntireX)** from the context menu.

To define a logical node name

- Select the tree node for the required node, choose **Rename** from the context menu or press F2, and replace the existing value by the following:

  ```
  LOGBROKER=node-name
  ```

  where *node-name* denotes the logical EntireX Broker name.

To remove a logical node name

- Select the tree node for the required node, choose **Rename** from the context menu or press F2, and replace the string `LOGBROKER=` by an EntireX Broker name.

To display physical names defined for logical services or nodes

- Use the ping command as described in the section *Server Command Execution*.

  Ping invokes a window that displays the physical node and server names defined for a logical service or a physical node name defined for a logical node.

Logon Option

If the **Logon Option** is set, for each `CALLNAT` request, the client initiates a Natural logon to the server using the current library name on the client, regardless of the library specified in the service directory. You can use the Application Programming Interface USR4008N to specify a different library; see also *Logging on to a Different Library* in *Using the Logon Option* in the *Natural Remote Procedure Call (RPC)* documentation.

After the remote `CALLNAT` has been executed (successfully or not), the server library is reset to its previous state. For more information, see *Using the Logon Option* in the *Natural Remote Procedure Call (RPC)* documentation.

The **Logon Option** can be set at server or node level and applies to all definitions made at a hierarchically lower level. If the **Logon Option** has been set for a certain server, it applies to all associated library and subprogram definitions.

To set the Logon Option

1. In the **Service Directory** tree view, select the tree node of a node, a server or a logical service and choose **Logon Option** from the context menu.

2. Choose **Yes** to set the **Logon Option** for the server. (The default is **No**.)

   If the **Logon Option** has been set successfully for the node selected, the icon indicating a node changes from to .

   If the **Logon Option** has been set successfully for a logical service, the icon indicating a logical service changes from to .

   If the **Logon Option** has been set successfully for the server selected, the icon indicating a server changes from to .

Transport Protocol

To specify the transport protocol

- In the **Service Directory** tree view, select the tree node of a node, a server or a logical service and choose **Transport Protocol** and **ACI** for EntireX Broker ACI protocol from the context menu.