SAProuter (BC-CST-NI)

HELP.BCCSTROUT

Release 4.6C

Copyright

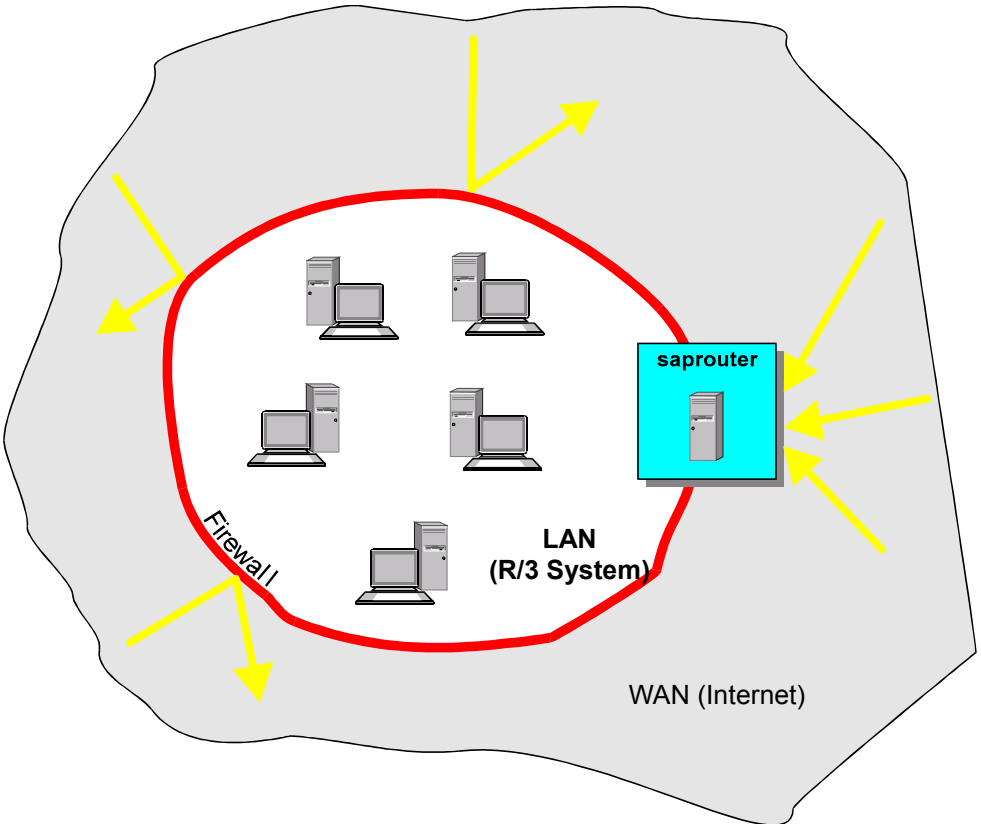
Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax
	Tip

Contents

SAProuter (BC-CST-NI)

SAProuter (BC-CST-NI)



What is SAProuter?

Purpose

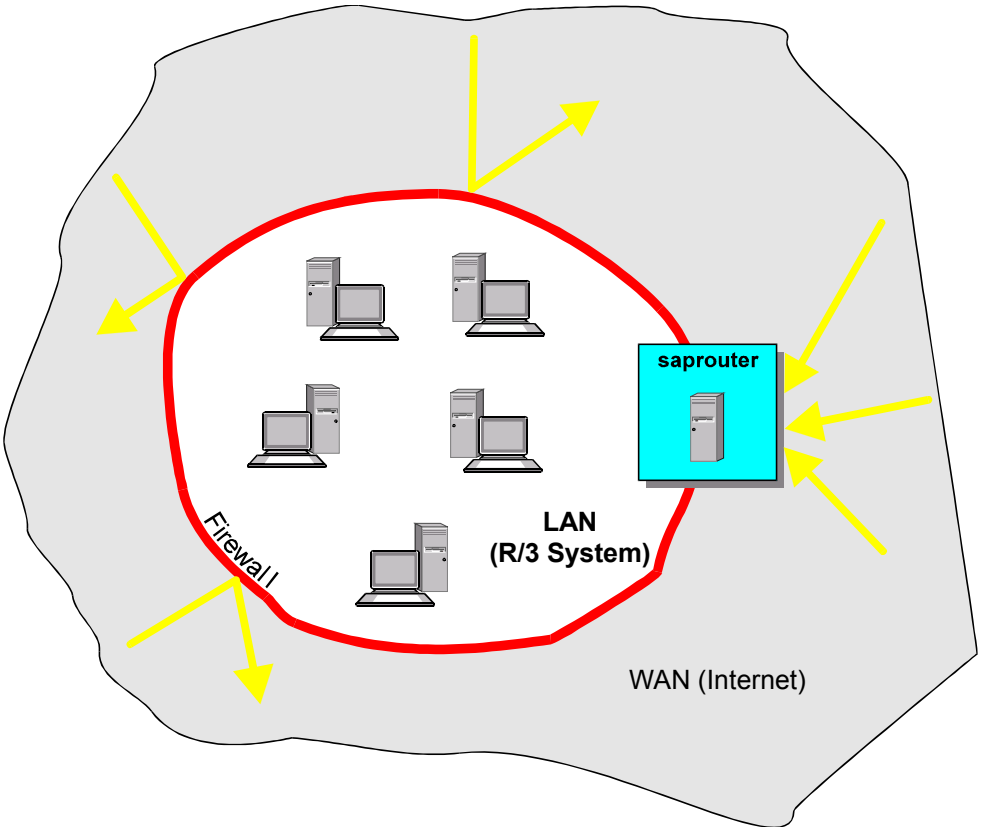
SAProuter is an SAP program that serves as an intermediate station (proxy) in a network connection between R/3 Systems or programs. It controls access to your network (application level gateway): This makes it a useful extension to an existing firewall system (port filter). A firewall forms a secure "wall" around your network. However, some connections have to be allowed through the wall. This means that the wall has to have a "hole". SAProuter controls this hole and gives you complete control over access to your R/3 System.

You can use SAProuter to:

- Control and log the connections to your R/3 System, e.g. from an SAP service center
- Set up an indirect connection when programs involved in the connection cannot communicate with each other due to the network configuration.
    - Address conflicts when using non-registered IP addresses
    - Restrictions which exist for firewall systems
- Improve network security by
    - Protecting your connection and data from unauthorized external access with a password
    - Allowing access from only particular SAProuters
    - Only allowing encrypted connections from a known partner (using the SNC layer)
    - Increase performance and stability by reducing the R/3 System load within a local area network (LAN) when communicating with a wide area network (WAN).

The following graphic illustrates your network (LAN) using a firewall as protection against access from outside. There is SAProuter running on the firewall host serving as “door” to your network. This door is only opened for connections you allow.

What is SAProuter?



Implementation Considerations

It is often useful if there is a connection from SAP to your R/3 System. In the SAPNet R/3 frontend (previously referred to as OSS), you can see important information and notes and SAP employees can log onto your system if there are problems, etc. These connections are controlled with SAProuter.

Integration

Note that you cannot protect your network from external access if SAProuter is installed without a firewall. You must ensure that all incoming connections go through the SAProuter "hole".

See also:

Using SAProuter [Page 21]

SAP Network Connections

The following describes the network connections for SAP Systems, the role played by SAProuter, and how you can increase your network security with SAProuter.

NI Network Interface [Page 10]

SNC - Secure Network Communication [Page 12]

Route Connections [Page 13]

Network Security with SAProuter [Page 15]

NI Network Interface

Definition

To provide independence from the various platforms, SAP has developed the intermediate layer **NI** (**N**etwork **I**nterface) for all network connections. It is used by SAProuter and all R/3 programs, as well as by the development kits for CPI-C and Remote Function Call (RFC).

Structure

In the OSI 7 layer model, the NI layer forms the upper part of the transport layer, and is therefore the part nearer the applications. Specifically, this means that NI uses TCP or UDP. The protocol is also known as the SAP Protocol [Page 11] .

NI in the OSI 7 layer model

OSI layer	Protocol
7 Application	
6 Presentation	
5 Session	
4 Transport	**NI** TCP / UDP
3 Network	IP
2 Data Link	Ethernet,...
1 Transfer method	

The test program `niping`, which tests the NI functions, belongs to the NI layer. A predefined number of data packages is simply sent from the client to the server, is returned by the server, and read again by the client. The program also outputs average transfer times and - depending on the trace level - detailed information on the data transfer. `niping` can be used to test network connections with or without SAProuter.

If **`niping`** is entered without parameters, an online help is displayed with possible parameters and additional options.

See also:

Testing SAProuter Basic Functions [Page 23]

SAP Protocol

Definition

The protocol used by SAP programs that communicate using the NI interface is called the **SAP Protocol**. This is an enhanced version of the TCP/IP protocol, which has been supplemented by one length field and some options for error information .

Use

When defining the Route Permission Table [Page 32], you can use **s** as the initial letter. This will then only allow SAP protocol, that is, the line will be interpreted as usual, but in addition only SAP programs (GUIs, servers) will be permitted to communicate with each other.

Integration

The NI network interface provides the SAP protocol as the default for communication, although it can also use the TCP/IP protocol with external programs (for example, `telnet` or `lpd`) that do not 'speak' SAP protocol.

SNC - Secure Network Communication

Use

SNC is used to make network connections using the Internet, in particular WAN connections, secure. It provides reliable authentication as well as encryption of the data to be transferred.

SAProuter allows SNC connections to be set up. The route permission table can be used to specify precisely whether and which SNC connections are allowed.

Prerequisites

At least SAProuter Version 30 and SNC configuration according to the appropriate guide.

To be able to set up a SNC connection between two SAProuters

- Each of the two SAProuters must have been started with the option **`-K [<SNCname>]`** (AS/400: **`-K <SNCname>`**) (see Option -K <mysncname> [Page 46]). These names ensure the authenticity of a host.
- There must be a **`KT`** entry in the route permission table of the source host, causing the connection to the target host to use the SNC layer.
- There must be a **`KP`** entry allowing the connection in both route permission tables.

Activities

To set up a SNC connection between two SAProuters, you must start them with the option **`-K`** and configure the Route-Permission-Tabelle [Page 28] appropriately.

See also:

Route Connections [Page 13]

Route Connections

Definition

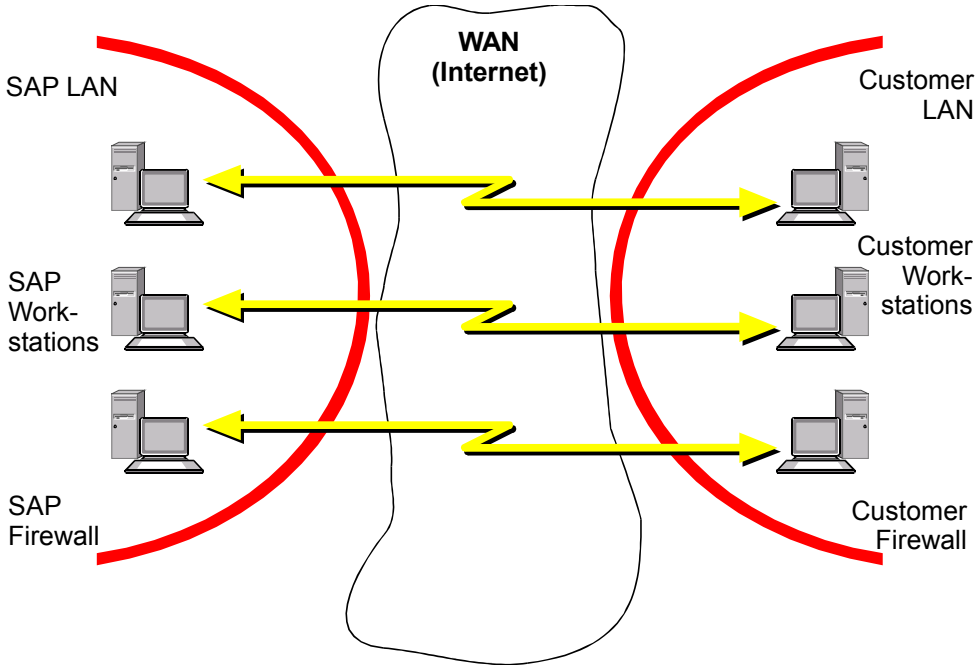
A route connection is a connection between two hosts using a network; the route is a sequence of intermediate stations used to set up the connection.

Structure

You can set up a connection between R/3 Systems with or without SAProuter.

Connections Without SAProuter

The following graphic shows a network connection without SAProuter.



We are assuming that both the SAP LAN (local area network) as well as the customer LAN are protected against unwanted access by firewalls.

If a connection is to be set up between an SAP workstation and a customer workstation, a “hole” needs to be made in the firewall; the more connections required to external hosts, the more holes (and therefore security gaps) the firewall contains.

If a connection is set up without SAProuter, the following information is required:

1. IP address of the host

   Or the logical name of the host on which the server process is running. The target host must therefore have a unique IP address.

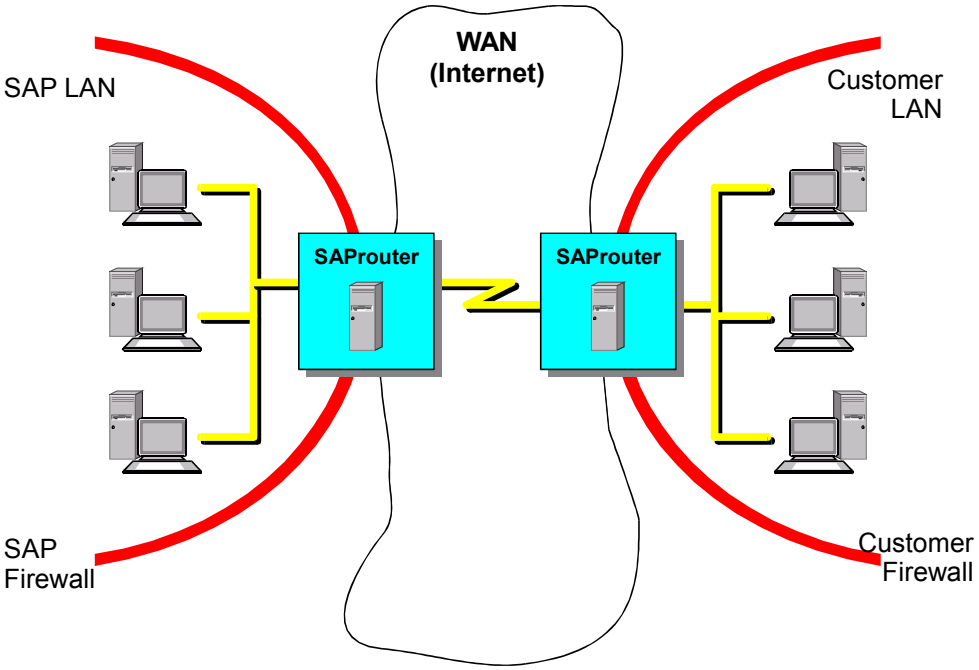
2. Port number or the logical name of the port used by the process

   The server process must use an exclusive port number on its host. This port number must be known to the client.

When the NI network interface [Page 10] is used, the host address and port number can be passed as logical names (for example, host **saposs**, service **sapdp00**) or address strings (for example, a host IP address in the form **www.xxx.yyy.zzz**, port **3200**).

Connections with SAProuter

The following graphic shows a network connection with SAProuter:



SAProuter only allows a network to be accessed from fixed points. The number of access points (“holes”) is therefore reduced, since fewer direct lines are required for connections. Each SAProuter has its own Route Permission Table [Page 28], which determines which routes can be used and which passwords are required for access. The hole in the firewall is therefore monitored.

Without SAProuter, the IP addresses must be unique. This is not always possible, particularly in the case of a connection between two networks which do not normally have an external connection. SAProuter enables two points with identical IP addresses to be connected.

SAProuter cannot only be used to connect one host with a particular service, but also several hosts and services with each other. The route information is provided in the form of a Route String [Page 25]. The passwords required for access are also specified in the route string.

See also:

Using SAProuter [Page 21]

Network Security with SAProuter

Purpose

SAProuter provides many functions designed to increase security, such as using SNC - Secure Network Communication [Page 12].

SAProuter allows you to strengthen you firewall host against unwanted connections from outside, if you administer it appropriately.

Implementation Considerations

Particularly the SAProuter running on your firewall host should be configured so that:

- Only the NI protocol (SAP Protocol [Page 11]) is accepted from outside, unless a native TCP/IP connection is explicitly required for a special port; use the `S` entries for this.
    - Not any number of SAProuters are allowed as stations before and after this SAProuter in a route; you can set this with the `Pv,n` entries.

Under UNIX, the SAProuter can also be started using Option -S <service> [Page 50] on a port reserved for root.

Installation of SAProuter

The following describes how to install SAProuter. Under UNIX, SAProuter is installed as a daemon, under Windows NT as a service.

Installation under UNIX [Page 17]

Installation under Windows NT [Page 18]

Installation under OS/400 [Page 19]

Installation under UNIX

1. Create the subdirectory `saprouter` in the directory `/usr/sap/`.
2. Get the most recent version of the SAProuter from sapserv3, directory `/general/misc/saprouter/`. Please refer to the related file `README` in this directory. Copy the programs `saprouter` and `niping` to the newly created directory `/usr/sap/saprouter`.

   If you cannot copy the programs from `sapserv3`, you can copy a version (may be obsolete) from your directory `/usr/sap/<SID>/SYS/exe/run`.
3. In file `/users/<SID>adm/startsap_<hostname>_<instance number>`, enter the following lines:

```
#
# Start saprouter
#
SRDIR=/usr/sap/saprouter
if  -f SRDIR/saprouter  ; then
         echo "\nStarting saprouter Daemon " | tee -a $LOGFILE
         echo "----------------------------" | tee -a $LOGFILE
$SRDIR/saprouter -r -W 30000 -R $SRDIR/saprouttab  \
               | tee -a $LOGFILE &
fi
```

4. Maintain the route permission table [Page 28] in directory `/usr/sap/saprouter`. If you want to keep it in another directory or under a name other than `saprouttab`, you must specify this with the SAProuter option **-R** (see Option R <routtab> [Page 45]).

Installation under Windows NT

Prerequisites

The SAProuter version must not be under 23.

Procedure

1. Create the subdirectory `saprouter` in the directory `<drive>:\usr\sap`.
2. Get the most recent version of the SAProuter from sapserv3, directory `/general/misc/saprouter/`. Please refer to the related file `README` in this directory. Copy the executables `saprouter.exe` and `niping.exe` to the directory you have just created.

   If there is no SAProuter there, you can get a version (may be obsolete) from your directory `<drive>:\usr\sap\<SID>\SYS\exe\run`.
3. If SAProuter has already been entered as a service with `srvany.exe`, remove the definition of the service from the Registry and restart the host.

   Define the service with the following command:

   `ntscmgr install SAProuter -b...\saprouter\saprouter.exe -p "service -r <parameter>"`

   Please note:

   The points stand for **`<drive>:\usr\sap`**

   `<parameter>` can be replaced by other parameters with which SAProuter should be started. It is important that the parameters are within the character string enclosed in double quotation marks.
4. Define the general attributes of the service: In *Control Panel* → *Services*, set the startup type to “automatic” and enter a user. SAProuter should **not** run under the SystemAccount.
5. To avoid the error message “The description for Event ID (0)” in the Windows NT event log, you must enter the following in the registry: Under *HKEY_LOCAL_MACHINE* → *SYSTEM* → *CurrentControlSet* → *Services* → *Eventlog* → *Application*, create the key `saprouter` and define the following values under it:

   EventMessageFile (REG_SZ):**`...\saprouter\saprouter.exe`**

   TypesSupported (REG_DWORD): **`0x7`**

   These adjustments are not obligatory for running SAProuter. They only provide detailed error messages in the event log.

Maintain the Route Permission Table [Page 28] in the `system32` directory of Windows NT. If you want to keep it in another directory or under a name other than `saprouttab`, you must specify this with the SAProuter option **`-R`** (see Option R <routtab> [Page 45]).

Installation under OS/400

Prerequisites

You have fetched the newest version of SAProuter from sapservX from the directory `general/misc/saprouter`, and you have read the corresponding README file.

Procedure

Import the programs SAPROUTER and NIPING into a separate library (for example, SAPROUTER).

1. Log on as `<SID>OFR`.
2. Create a library:

   ```
   CRTLIB <libraryname>
   ```

3. Create a backup file SAPROUTER:

   ```
   CRTSAVF <libraryname>/SAPROUTER
   ```

4. Create a backup file NIPING:

   ```
   CRTSAVF <libraryname>/NIPING
   ```

5. Import the programs SAPROUTER and NIPING with ftp:

   ```
   ftp sapservX
   cd general/misc/saprouter
   lcd SAPROUTER
   bin
   get saprouter<.version><.platform> SAPROUTER (replace
   get niping<.version><.platform> NIPING (replace
   quit
   ```

6. Recover the SAPROUTER objects:

   ```
   RSTOBJ OBJ(*ALL) SAVLIB(SAPROUTER) DEV(*SAVF)
   SAVF(SAPROUTER/SAPROUTER) RSTLIB(SAPROUTER)
   ```

7. Recover the NIPING objects:

   ```
   RSTOBJ OBJ(*ALL) SAVLIB(SAPROUTER) DEV(*SAVF)
   SAVF(SAPROUTER/NIPING) RSTLIB(SAPROUTER)
   ```

8. Create the directory `/usr/sap/saprouter`.
9. You must maintain the corresponding routing table under `/usr/sap/saprouter/saprouttab`. You can find an example of a routing table on sapservX in the aforementioned directory.

More information

Starting SAProuter [Page 22]

Using SAProuter

This chapter describes how SAProuter is started, tested, and configured.

Starting SAProuter

Prerequisites

Before using SAProuter, you should test its basic functions.

Testing SAProuter's Basic Functions [Page 23]

Procedure

To start SAProuter:

Enter **`saprouter -r`** in the input field ein (AS/400: Enter **`saprouter '-r'`** in the input field, if possible in batch mode).

This command starts SAProuter. The connections allowed are contained in the Route Permission Table [Page 28] `saprouttab`

You can start SAProuter automatically when booting the system. Under UNIX, for example, you change your file `/etc/rc` appropriately.

Main SAProuter commands and what they do:

`saprouter`	Displays a complete list of SAProuter parameters on the screen
`saprouter -r` (AS/400: **`saprouter '-r'`**)	Starts SAProuter
`saprouter -s` (AS/400: **`saprouter '-s'`**)	Stops SAProuter

See also:

SAProuter Options [Page 33]

Testing SAProuter's Basic Functions

Prerequisites

Before using SAProuter, you should test whether there are any network problems.

You require the programs `saprouter` and `niping` as well as three open windows (shells) on one or more hosts.

Procedure

The following table shows the test scenario when using `niping`:

SAProuter runs in window 1, the server in window 2, and the client in window 3.

UNIX/NT

	Window 2 (host2)	Window 1 (host1)	Window 3 (host3)
Without SAProuter	`niping -s`		`niping -c -H host2`
With SAProuter	`niping -s`	`saprouter -r`	`niping -c -H /H/host1/H/host2`

AS/400

	Window 2 (host2)	Window 1 (host1)	Window 3 (host3)
Without SAProuter	`call niping '-s'`		`call niping '-c' '-H' 'host2'`
With SAProuter	`call niping '-s'`	`saprouter '-r'`	`call niping '-c' '-H' '/H/host1/H/host2'`

Steps

1. Start SAProuter in window 1 (on `host1`). To do this, enter the following command:

   UNIX/NT: **`saprouter -r`**

   AS/400: **`saprouter '-r'`**

   This command calls SAProuter without any parameters.

   For a complete list of the SAProuter commands, refer to the chapter SAProuter Options [Page 33] or the online help. To call the online help, enter **`saprouter`**.

2. In window 2 (`host2`), start the test program `niping` to emulate a test server. Enter the following command:

   UNIX/NT: **`niping -s`**

   AS/400: **`call niping '-s'`**

   For a complete list of the `niping` commands, refer to the online help. To call the online help, enter **`niping`**.

3. In window 3 (`host3`), start the test program `niping` to emulate a client. Enter the following command:

Testing SAProuter's Basic Functions

UNIX/NT: **`niping -c -H host2`**

AS/400: **`call niping '-c' '-H' 'host2'`**

This command tests the connection **without** SAProuter, that is directly between `host2` and `host3`.

4. In window 3, start the test program `niping` again with the following command:

   UNIX/NT: **`niping -c -H /H/host1/H/host2`**

   AS/400: **`call niping '-c' '-H' '/H/host1/H/host2'`**

   This command tests the connection **with** SAProuter. A host name is interpreted as a route (over one or more SAProuters to the server) if **`/H/`** is added as a prefix to the host name (see Route Strings [Page 25]).

In steps 3 and 4, data packages are sent to the server, and the server sends the data packages back. In step 3, the data packages should be sent to the server more frequently, since more process changes take place.

To perform a self test for the local host:

Enter the command **`niping -t`** (AS/400: **`call niping '-t'`**).

A list with function names, parameters, and return codes is displayed. If the self test is successful, the following message appears:

```
*** SELFTEST O.K. ***
```

To get an idea of the options provided by `niping`, enter **`niping`** without any parameters.

See also:

Route String Entry for SAProuter [Page 26]

NI Network Interface [Page 10]

Route Strings

Definition

A route string describes the stations of a connection required between two hosts. A route string has the syntax

`(/H/host/S/service/W/pass)*`

It consists of any number of **substrings** in the form `/H/host/S/service/W/pass`.

H, **S**, and **W** must be uppercase!

Structure

A route string contains a substring for each SAProuter and for the target server.

Each substring contains the information required by SAProuter to set up a connection in the route: the host name, the port name, and the password, if one was given.

Syntax for substrings:

- `/H/` indicates the host name
- `/S/` is used for specifying the service (port); it is an optional entry, the default value is `3299`
- `/W/` indicates the password for the connection between the predecessor and successor on the route and is also optional (default is “”, no password)

  In earlier Releases (<4.0A), the password entry was made one substring later and with the letter `/P/`.

  New: `/H/saprouter/W/pass/H/targetserver`

  Old: `/H/saprouter/H/targetserver/P/pass`

  (Here `pass` is the password which is checked by the SAProuter on host `saprouter` to set up or prohibit the connection from the source host to the target host.)

  Due to downward compatibility, the old password entry form is still possible.

See also:

Route String Entry for SAProuter [Page 26]

Route String Entry for SAProuter

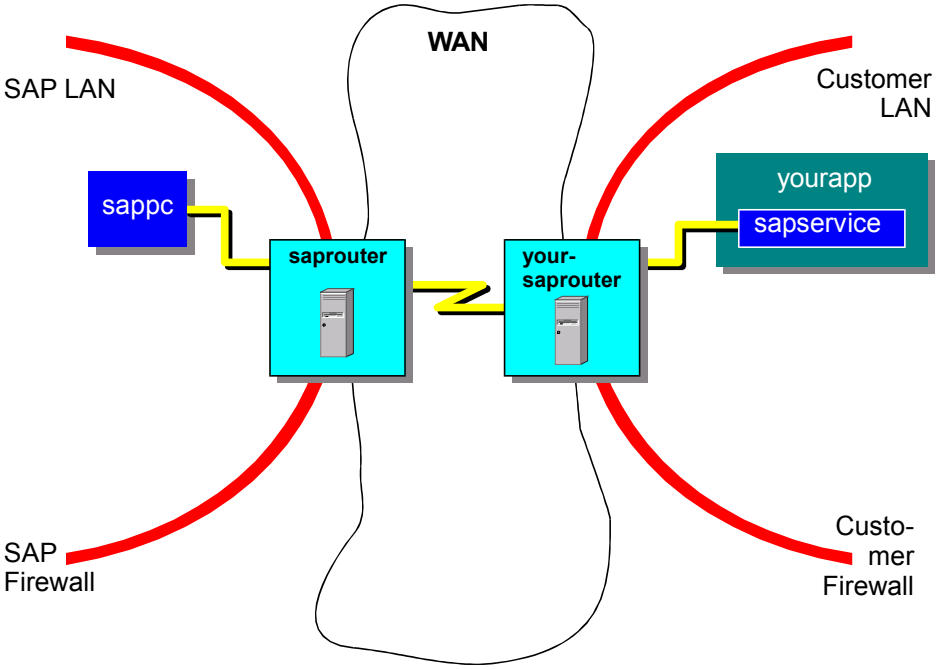
Purpose

A route string describes a connection required between two hosts using one or more SAProuters. Each of these SAProuters then checks its Route Permission Table [Page 28] to see whether the connection between its predecessor and successor is allowed, and if it is, sets it up.

Process Flow

The entry of route strings is best illustrated by an example.

The following graphic shows an example of a connection between SAP and a customer system. In this example, an SAP employee working on `sappc` wants to log on to a customer application server `yourapp`, which provides or uses the service `sapservice`.



The SAP service employee logs on to the R/3 System and sets up a connection between `sappc` and `yourapp` using the SAProuter on `saprouter` and the customer's SAProuter `yoursaprouter`.

`yoursaprouter` requires the password `pass_to_app` for connections with `yourapp`.

The route string appears as follows:

`/H/saprouter/H/yoursaprouter/W/pass_to_app/H/yourapp/S/sapservice`

This route string is interpreted by the SAProuters involved in the route as follows:

	Host/Address	Service/Port	Password

Substring 1	**/H/saprouter**	**/S/<default>**	**<no password>**
Substring 2	**/H/yoursaprouter**	**/S/<default>**	**/W/pass_to_app**
Substring 3	**/H/yourapp**	**/S/sapservice**	

The connection from **sappc** to the application server is set up in the following steps:

sappc (frontend)	Sets up the connection to SAProuter saprouter according to substring 1 and relays the route information.
saprouter (SAProuter)	Uses the Route Permission Table [Page 28] to check whether the route "sappc to yoursaprouter 3299" is allowed, sets up the connection to the SAProuter on yoursaprouter, and passes substring 2 and 3.
yoursaprouter (SAProuter)	Checks whether the route "saprouter to yourapp, sapservice" is allowed. The password pass_to_app is also checked. SAProuter then sets up the connection to the application server.

A SAProuter always checks only the **previous** host name or the previous IP address and the **next** substring (**/H/.../S/.../W/...**) for host name or IP address, service and password. The last substring does not contain a password, since there is no successor in the route.

If the **/S/** section is missing, the default port number of the SAProuter is used. If the **/W/** section is missing, a password is not used.

With the old password entry, the above route string would appear as follows:

/H/saprouter/H/yoursaprouter/H/yourapp/S/sapservice/P/pass_to_app

See also:

Route Strings [Page 25]

Route Permission Table [Page 28]

Route Permission Table

Definition

The route permission table contains the host names and port numbers of the predecessor and successor points on the route (from the SAProuter's point of view), as well as the passwords required to set up the connection (corresponds to a substring, cf. Route Strings [Page 25]). It is used to specify which connections are allowed and which prohibited by SAProuter. It also specifies whether SNC connections are set up and which these are.

Structure

Standard Entries

Standard entries in a route permission table appear as follows:

P/S/D <source-host> <dest-host> <dest-serv> <password>

<source-host> and <dest-host> could be SAProuters.

The beginning of the line can be as follows:

- **P**(ermit) causes SAProuter to set up the connection. **P**(ermit) entries can contain a password. SAProuter checks whether this password corresponds to that sent by the client.

  Directly after the **P**, you can also specify the maximum number of SAProuters permitted before and after this SAProuter on the route for the connection to be allowed: **Pv,n** – here **v** denotes the maximum number of preceding SAProuters on the route, **n** the maximum number of following ones.

- **S**(ecure) only allows connections with the SAP Protocol [Page 11]; connections with other protocols (such as TCP) are not allowed, see Network Security with SAProuter [Page 15].

- **D**(eny) prevents the connection from being set up.

- You can also add comment lines, which must begin with '**#**'.

If a <source-host> client wants to set up a connection to <dest-host> <dest-serv> using SAProuter, SAProuter checks its route permission before the connection is set up. If the password and route SAProuter has received correspond to the entries in the route permission table, SAProuter sets up the connection. Otherwise, SAProuter does not set up the connection.

A route permission table could appear as follows:

D	**host1**	**host2**	**serviceX**	
D	**host3**			
P	*****	*****	**serviceX**	
P	**155.56.*.***	**155.56**		
P	**155.57.1011xxxx.***			

P	host4	host5	*	pass
S	host6			
P	host7	host8	telnet	
P*,0	*	*		gui

This means:

- Do not allow any routes from `host1` to `host2`, service `serviceX`
- Do not allow any routes starting from `host3`
- Allow all routes to server processes using `serviceX`
- Allow all routes within subnetwork 155.56
- Allow all routes starting from subnetwork 155.57.1011xxxx (the last byte is written as a binary number, each “x” stands for 0 or 1)
- Allow all routes from `host4` to `host5` if password `pass` is correct
- All routes from `host6`, but only SAP protocol
- Native protocol routes (TCP/IP) from `host7` to the non-SAP service telnet on `host8`
- All connections to non-SAProuters (no more SAProuters allowed on this route) if password `gui` is correct

In the above example in Route String Entry for SAProuter [Page 26] the route permission table of host `saprouter` must have the entry

```
P  sappc  yoursaprouter
```

and the route permission table of host `yoursaprouter` must contain the entry

```
P  saprouter  yourapp  sapservice  pass_to_app
```

as well.

First Match

The first entry in the route permission table for which source address, target address, and target port match is decisive; in the above example, this means that the connection from `host1` to `host2`, service `serviceX` is not allowed (because of the first entry), although all connections with service `serviceX` are allowed according to the third entry.

Exception

If the SAProuter is the last SAProuter on the route (followed e.g. by the frontend) and the service is not an SAP service (no SAP protocol), the wildcard (“*****”) cannot be used with the service. The connection is only allowed if the non-SAP service is selected explicitly; if the example given above contained a ***** instead of **telnet** and the SAProuter was the last one on the route, the telnet connection would not be set up.

SNC Entries

SNC entries always start with the letter **K** (like **k**ey).

There are two types of SNC entries:

1. **KT** entries (**K**ey **T**arget)

   This defines which connections should be SNC connections. This can be defined for both incoming and outgoing connections (from the point of view of this SAProuter).

   a) Incoming connections

   The syntax is **KT <SNCname src-host> <src-host> <src-serv>**.

   This means that connections coming from the host **<src-host> <src-serv>** with the SNC name **<SNCname src-host>** should be SNC connections.

   The user can thus define that service connections from SAP must be SNC connections.

   b) Outgoing connections

   They have the syntax **KT <SNCname dest-host> <dest-host> <dest-serv>**. This means that connections from the SAProuter to **<dest-host> <dest-serv>** with the SNC name **<SNCname>** should be SNC connections.

   So that SNC connections are possible, the appropriate SAProuters need to have been started with the option **-K and** the route permission table must contain the appropriate **KT** entry!

2. **KD**, **KP**, and **KS** entries

   They have the following syntax:

   K<D/P/S> <SNCname source-host> <dest-host> <dest-serv> <password>. This means that an (encrypted) SNC connection from **<SNCname source-host>** via SAProuter to **<dest-host> <dest-serv>** is set up when the route string contains the correct **<password>**.

P	*	*	*	pass
KT	S:SR@host4	host4	3333	
KT	S:SR@host4	host9	*	
KD	S:SR@host4	host9	*	
KP	S:SR@host4	*	*	pass2
KS	*	host10	4444	
KP	*	*	*	

This means:

- Allow all connections if password `pass` is specified correctly
- Connections from `host4` (SNC name `S:SR@host4`), service 3333 to this SAProuter should be SNC connections
- Connections from this SAProuter to `host9` (SNC name `S:SR@host9`) should be SNC connections
- A SNC connection from `SR@host4` to `host9` using this SAProuter should **not** be set up
- A SNC connection from `S:SR@host4` using this SAProuter (any target host) is allowed if the password `pass2` is correct (unless the connection is to `host9`, since this is not allowed according to the previous entry - the first entry which “matches” is decisive!)
- All SAP-SAP connections (that is NI protocols) to `host10`, service `4444` which enter as SNC connections are passed on to `host10` (no SNC host) as non-SNC connections.
- All SNC connections (for which the previous entries are not suitable) are allowed.

Creating a Route Permission Table

You can create a route permission table with a standard text editor.

- You must create a separate route permission table for each SAProuter in your network.
- If a specific route permission table has not been assigned to the SAProuter, `./saprouttab` is used under UNIX, and under Windows NT the file `saprouttab` in the working directory of the SAProuter `<lwk>:\usr\sap\saprouter is searched for`. If this file is not available, SAProuter terminates with an appropriate message.

You can use generic entries ("**`*`**") in hosts, ports, and passwords.

You can use sub-networks in host routes. Examples:

`156.56.*.*`	All host addresses beginning with **`156.56`**
`133.27.17.*`	All host addresses beginning with **`133.27.17`**
`156.56.1011xxxx`	All host addresses from **`156.56.176.*`** to **`156.56.191.*`**. (Binary interpretation of the third byte of the address. '**`x`**' is a freely selectable binary value (**`1`** or **`0`**).)

You can display an example of a route permission table on the screen. To do this, call the SAProuter online help: **`saprouter`**.

See also:

Route Permission Table [Page 28]

Route String Entry for SAProuter [Page 26]

SAProuter Options

SAProuter provides some functions that can be used optionally. They consist of a letter, which is specified when SAProuter is called (UNIX/NT syntax: **`saprouter -<option>`**>, AS/400 syntax: **`saprouter -'<option>'`**) or which is sent to a running SAProuter.Its use and default values are described below.

There are **administrative options** (lowercase), **additional options**, and **expert options** (uppercase). The various options can be combined, provided this makes sense, by specifying an administrative option and any number of other options:

UNIX/NT: **`saprouter [-<adm>] [-<opt>]`**
AS/400: **`saprouter '[-<adm>] [-<opt>]'`**

If an invalid combination of SAProuter options is specified, SAProuter behaves as if only **`saprouter`** was specified and shows the online help.

Administrative Options [Page 34]

Additional Options [Page 44]

Expert Options [Page 57]

Administrative Options

Purpose

Administrative options — with the exception of the startup functions **`-r`** and **`-a <lib>`** — are sent to a running SAProuter, which then executes the appropriate function.

SAProuter is started with the command **`saprouter -r`** (AS/400: **`saprouter '-r'`**), see Starting SAProuter [Page 22].

Features

The following list provides an overview of the administrative options:

Options must be placed in quotation marks under OS/400, for example, enter **`saprouter '-s'`** to stop SAProuter.

Option -s (stop saprouter)

Use

This function is used to stop a running SAProuter.

Integration

If the SAProuter to be stopped is not running on the default service `3299`, the service has to be made known with option -S <service> [Page 50].

The commands **`saprouter -s -S 3299`** and **`saprouter -s`** (AS/400: saprouter '-s -S 3299' and saprouter '-s') are equivalent.

Option -n (new saprouttab)

Use

The command **saprouter -n** (AS/400: **saprouter '-n'**) is used to report changes in the route permission table to the running SAProuter. It causes SAProuter to use the updated table, as named with option -R <routtab> [Page 45] (default `saprouttab`).

If you would like to enter, for example, other restrictions in the route permission table, you do not have to stop and restart SAProuter, but you can use this function.

The new route permission table does not affect connections which already exist! Even if the existing connection is not allowed according to the new table, it is retained!

Option -t (toggle trace)

Use

This function is used to toggle the trace level of a running SAProuter. Trace levels 1, 2 and 3 exist. If the trace level was 1, it is now increased to 2, and if it was 2 or 3, it is decreased to 1.

Integration

When SAProuter is started, the trace level is selected with option -V<tracelev> [Page 49].

Option -c<n> (cancel connection n)

Use

Internally, each connection using SAProuter has a number, which can be seen with option -l / -L [Page 39]. This function can be used to close a connection.

The command **`saprouter -c 2`** (AS/400: **`saprouter '-c 2'`**) closes the connection with the (internal) number 2.

Option -I / -L

Use

You can use the **`saprouter -l`** (AS/400: **`saprouter '-l'`**) function if you want SAProuter to display route information on the screen. The **`saprouter -L`** (AS/400: **`saprouter '-L'`**) function provides even more detailed information.

The information contains:

- A table with the connection number, client, partner, and service for each existing connection
- The total number of clients, the working directory in which SAProuter is running, and the path of the route permission table [Page 28].

If you want to display the SAProuter information from a remote host, you should use the option -H <hostname> [-P <kennwort>] [Page 53].

Option -d (dump buffers)

Use

If this function is used, detailed information on the host names involved in the connection and their IP addresses is written to the trace file (default **dev_rout**, or the name specified with option -T<tracefile> [Page 48]). The trace file is not overwritten, the information is simply appended at the end.

Option -f (flush buffers)

Use

This function can be used to empty the internal buffer (which is written to the trace file with option -d (dump buffers) [Page 40]).

Option -p

Use

This option can be used to perform a soft shutdown of SAProuter. SAProuter continues running on another port, can be administered on this port, but does **not** accept any **logon requests**, and terminates automatically when there are no more clients connected.

The port on which SAProuter was running before (default 3299) is now free. This is useful if:

- A new SAProuter is to be started without closing all existing connections
- More connections are required than one SAProuter alone can provide (max. 1018).

If you enter the command **`saprouter -p`**, information is displayed telling you on which port SAProuter can now be administered, and the host on which SAProuter is running.

The standard port on which SAProuter is running is port 65000. If it is already assigned or if a port range was already defined for the SAProuter with option -M <min>.<max> [Page 56], a different port is selected.

Option -a <lib>

Use

This option is **not** sent to a running SAProuter, but is used to **start** SAProuter with an external library. **<lib>** is the relative path name of the library. A string can also be passed to the library with option -A <initstring> [Page 55].

Note that SAP cannot guarantee support if you use an external library. Please contact the vendor of the external library if you have problems.

Additional Options

Purpose

The additional options — with one exception — are indicated by uppercase letters. They can be combined with each other and with an administrative option, as long as this makes sense. The ways in which the options can be combined are indicated in the sections in which they are described.

If an invalid combination of SAProuter options is specified, SAProuter behaves as if only **`saprouter`** was specified and shows the online help.

Implementation Considerations

The additional options can also be omitted, there are default values that are specified for each option.

Features

Option -R <routtab> [Page 45]

Option -K <mysncname> [Page 46]

Option -G<logfile> [Page 47]

Option -T<tracefile> [Page 48]

Option -V<tracelev> [Page 49]

Option -S <service> [Page 50]

Option -C <clients> [Page 51]

Option -H <hostname> [-P <password>] [Page 53]

Option -A <initstring> [Page 55]

Option -M <min>.<max> [Page 56]

Options must be placed in quotation marks under OS/400, for example, enter **`saprouter '-s'`** to stop SAProuter.

Option -R <routtab>

Use

You can use the **`saprouter -R <path>`** (AS/400: **`saprouter '-R <path>'`**) option to specify the file containing the route permission table. If an entry is not made, SAProuter searches the file

- `./saprouttab` (UNIX and AS/400)
- `<lwk>:\usr\sap\saprouter\saprouttab` (NT)

The route permission table is essential for SAProuter (version >= 23). If it is not found, SAProuter terminates with an appropriate message.

If you want to permit all connections, you must specify the following single-line route permission table:

```
P * * *
```

Option -K <mysncname>

Use

For SNC connections to be possible with SAProuter, SAProuter must be started with this option: **`saprouter -r -K <mysncname>`** (AS/400: **`saprouter '-r -K <mysncname>'`**). There must also be a **`KT`** entry in the route permission table [Page 28] specifying that connections with a certain host (whose SNC name is known) should be SNC connections. `<mysncname>` is the SNC name of the host on which the SAProuter is running.

See also:

SNC - Secure Network Communication [Page 12]

Option -G<logfile>

Use

When you start your SAProuter, you can specify a log file.

UNIX/NT: **saprouter -r -G <logfile>**

AS/400: **saprouter '-r -G <logfile>'**

<logfile> is the name (relative path name) you specify for the log file. All important activities, such as starting the connection and runtime operations, are logged in this file:

- Connection from (client name/address)
- Connection to (partner name/address)
- Partner service
- Start time
- End time
- Connection requests rejected after checking the route permission table [Page 28].

If this option is not used, a log file is not created.

Option -T<tracefile>

Use

A trace file is used to search for and correct errors. It logs in detail - the higher the trace level (see Option -V<tracelev> [Page 49]), the more detailed the information - what SAProuter does. From this, you can see in which function an error occurred, why a connection was not established, etc.

When you start SAProuter, you can specify a trace file:

UNIX/NT: **`saprouter -r -T <tracefile>`**
AS/400: **`saprouter '-r -T <tracefile>'`**

A trace file always exists. If the option is not used, the trace file **dev_rout** in the working directory is used. It resides in the working directory of the SAProuter.

Option -V<tracelev>

Use

This option is used to set the trace level when SAProuter is started:

UNIX/NT: **`saprouter -r -V3`**

AS/400: **`saprouter '-r -V3'`**

for example, starts SAProuter with trace level 3.

The trace level specifies how detailed the information should be in the trace file: 1 means hardly any information, 3 very detailed information. The name of the trace file can be set with option -T<tracefile> [Page 48].

You can change the trace level while SAProuter is running with option -t (toggle trace) [Page 37].

Trace levels 1, 2, and 3 are available, and the default value is 1.

Option -S <service>

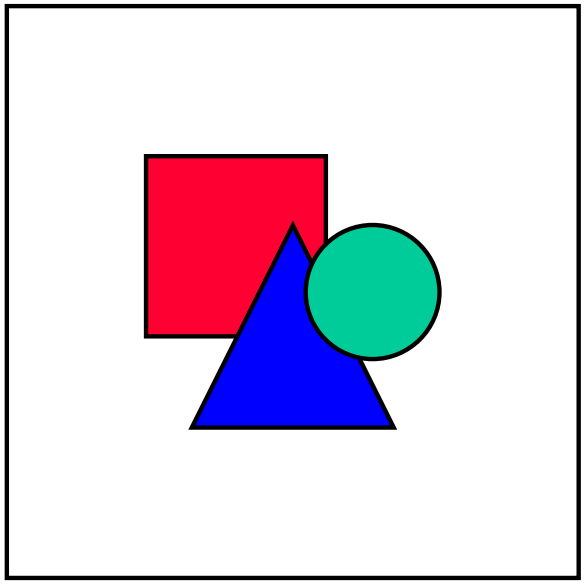
Use

The option **`-S <service>`** is used to specify the service (port) on which SAProuter runs (default 3299). SAProuter can, for example, be started on any other service: **`saprouter -r -S 4444`** (AS/400: **`saprouter '-r -S 4444'`**) starts SAProuter on the local host on service 4444. If you want to administer this SAProuter, of course you also have to define the service.

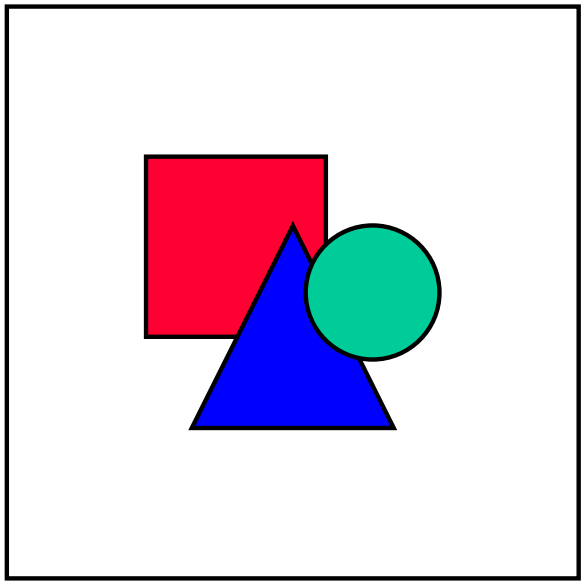
Option -C <clients>

Use

You can use this function to set the maximum number of clients. The default setting is 800, the maximum value is 2039.



Note that two clients correspond to one connection; that is max 400 connections are preset and max. 1019 connections are possible.



If you want to run 1000 connections with your SAProuter, start SAProuter as follows:

UNIX/NT: saprouter -r -C 2000
AS/400: **saprouter '-r -C 2000'**

Option -C <clients>

If you would like to have more connections than the maximum (1019), you can “move” SAProuter to another port with option -p [Page 42] and start a new SAProuter on this port.

These limitations are obviously only valid if smaller values for the number of connections have not been set in the operating system. Therefore you must take the operating system parameters into consideration.

Option -H <hostname> [-P <password>]

Use

This option has two uses:

1. You can define the option when you start SAProuter:

   saprouter -r -H <hostname> (AS/400: **saprouter '-r -H <hostname>'**).

   This means that SAProuter *only* responds to the IP address of host <hostname>; if option **-S** does not define any other value, this is default port 3299. If SAProuter is started without option **-H**, it responds to all IP addresses of this host. <hostname> can also be an IP address.

   The host myhost has two IP addresses: a1 and a2.

   The call **saprouter -r** (AS/400: **saprouter '-r'**) causes SAProuter to respond to a1/3299 and a2/3299. The call **saprouter -r -H a2** (AS/400: **saprouter '-r -H a2'**) causes SAProuter to respond only to a2/3299.

   If you started SAProuter with option **-H <hostname>**, you also have to define the host name for administration. For example, if you want to use a new route permission table, you must enter **saprouter -n -H <hostname>** (AS/400: **saprouter '-n -H <hostname>'**).

2. You can use this option in a running SAProuter to get SAProuter information (displayed with the option -I / -L [Page 39]) from a remote host. A password may be required, which is then entered with option **-P <password>** (AS/400: **Option '-P <password>'**). SAProuter then checks its route permission table [Page 28] to determine whether the route is allowed with this password, and if it is displays the information.

   SAProuter is running on host_sr, port 3299 (default). You would like to display the SAProuter information (list of all SAProuter clients, for example) from the host myhost.

   Enter the command **saprouter -l -H host_sr -P pass** (AS/400: saprouter '-l -H host_sr -P pass ').

   SAProuter checks whether its route permission table contains the entry

   P myhost host_sr 3299 pass

   or not, If it does, the SAProuter information is displayed on your host myhost.

Integration

If the SAProuter is running on a port other than the default port 3299, you can specify this in the command line with option -S <service> [Page 50].

Option -H <hostname> [-P <password>]

Option -A <initstring>

Use

This option is only required in connection with option -a <lib> [Page 43]. If SAProuter is started with an external library, another string can be passed to this library with option **`-A <initstring>`** (AS/400: Option **`'-A <initstring>'`**).

Option -M <min>.<max>

Use

You can use this option to specify a port range for outgoing connections. For example, the command **`saprouter -r -M 1.1023`** only allows outgoing connections from ports 1 to 1023 (reserved for root under UNIX).

Integration

This option can be used to increase security; see Network Security with SAProuter [Page 15]

Expert Options

Purpose

SAProuter has a few expert options, which are described below.

Please use these options only after consulting SAP or if you are very experienced in this area!

Features

Command	Function	Default
`-B <bufsize>`	Maximum queue length per client	500,000 bytes
`-Q <queuesize>`	Maximum total size of all queues	20,000,000 bytes
`-W <waittimeL>`	Timeout for blocking network calls (if there is an error)	5000 msec

Error Diagnosis

As a rule, always refer to the relevant notes in SAPNet if you experience problems with SAProuter.

Note number	Content
0012023	ERROR => NI_PONG in more than one package
0029684	STFK: Route permission denied
0062636	saprouter terminates on ending UNIX session
0063342	List: NI error codes
0139184	Saprouter: Invalid DATA from C...
0155839	SAProuter and the Year 2000
0163436	Check connection and raise a event when connect
0164937	NiPBind: service 'sap????' in use
0167857	niping -s error on Windows 95/8
0168937	AIX: Error code for accept exits server
0169398	Reliant: setup connection in the R/3 System fails
0180075	SAProuter for Linux
0181896	AS/400: Signal handling in NI
0184896	NI: Error correction NI
0104576	Package filter between ITS and R/3
0042692	Test tool for RFC links: sapinfo
0066168	Required documents when analyzing RFC problems
0025917	Changes to /etc/hosts are not accepted
0147021	"Address already in use" due to TCP state
0053459	SAP programs for Linux
0085749	Using SAProuter with SNC for secure printing
0037211	ftp not via SAProuter : "connection refused"

The error messages output directly by SAProuter are described under SAProuter Error Messages [Page 59].

SAProuter Error Messages

Definition

If an error occurs while SAProuter is in operation, an error message is displayed by the SAProuter client.

Structure

A SAProuter error message consists of eight or more lines, with a blank line inserted after one or two lines.

SAProuter error message

LOCATION	SapRouter on myhost
ERROR	partner not reached
TIME	Wed Jul 23 15:24:42 1997
RELEASE	40A
COMPONENT	NI (network interface)
VERSION	30
RC	-100
COUNTER	1

The first two lines are important. They indicate:

- On which host the SAProuter concerned is running (in this example `myhost`)
- To which application area the error belongs (here connection setup)

In this example, SAProuter cannot set up the connection to its partner. You are advised to check the connection again.

If there is no `LOCATION` entry, the error message refers to a local program.

The information after the blank line is particularly relevant for internal errors. If you cannot correct the error, the detailed information may be helpful when you contact SAP.

The most important error messages are:

Route permission denied [Page 60]

Maximum number of clients reached [Page 61]

Route permission denied

Prerequisites

One of the most common error messages is the following:

LOCATION	SapRouter on myhost
ERROR	route permission denied
TIME	.....
....	....

A connection has not been set up because SAProuter does not allow the route concerned.

Procedure

Check the route permission table [Page 28] **of this** SAProuter (on host `myhost`) carefully and change it, if necessary.

You can find out which working directory the running SAProuter and the route permission table are in with option -l / -L [Page 39].

Remember that the **first** entry in the route permission table for which **source address, target address,** and **target port** match is decisive!

You can import a modified route permission table with option -n (new saprouttab) [Page 36].

Maximum number of clients reached

Prerequisites

SAProuter does not accept a connection and outputs the following error message:

LOCATION	SapRouter on myhost
ERROR	maximum number of clients reached
TIME	.....
....	....

This means that SAProuter cannot accept any further clients because the maximum number has been reached (default 800). SAProuter continues execution with all other clients.

Procedure

In order not to have to restart SAProuter (and thereby end all existing connections), you should perform a soft shutdown of the SAProuter with option -p [Page 42]; SAProuter will continue running on a different port. SAProuter can then be started on the old port, possibly with a larger number of clients (see option -C <clients> [Page 51]). It will then accept clients again.