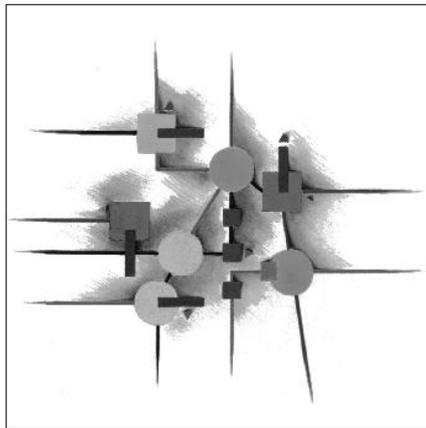


Secure Network Communications Test Plan



S Y S T E M R / 3

Version 1.05

Release 4.0



Copyright

©Copyright 1997 SAP AG. All rights reserved.

No part of this documentation may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG.

SAP AG further does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP AG shall not be liable for any special, indirect, incidental, or consequential damages, including without limitation, lost revenues or lost profits, which may result from the use of these materials. The information in this documentation is subject to change without notice and does not represent a commitment on the part of SAP AG in the future.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft®, WINDOWS®, NT® and EXCEL® and SQL-Server® are registered trademarks of Microsoft Corporation.

IBM®, OS/2®, DB2/6000®, AIX®, OS/400® and AS/400® are a registered trademark of IBM Corporation.

OSF/Motif® is a registered trademark of Open Software Foundation.

ORACLE® is a registered trademark of ORACLE Corporation, California, USA.

INFORMIX®-OnLine *for SAP* is a registered trademark of Informix Software Incorporated.

UNIX® and X/Open® are registered trademarks of SCO Santa Cruz Operation.

ADABAS® is a registered trademark of Software AG.

SECUDE® is a registered trademark of GMD-German National Research Center for Information Technology.

SAP®, R/2®, R/3®, RIVA®, ABAP/4®, SAPoffice®, SAPmail®, SAPaccess®, SAP-EDI®, SAP ArchiveLink®, SAP EarlyWatch®, SAP Business Workflow®, R/3 Retail® are registered trademarks of SAP AG.

SAP AG assumes no responsibility for errors or omissions in these materials.

All rights reserved.

1 Test Plan SNC (Secure Network Communications)	4
1.1 Test Objective	4
1.2 Test Environment	4
1.3 Test Scenario Overview	5
<hr/>	
2 The Test procedure	6
2.1 Preparing the environment.....	6
2.1.1 Application Server	6
2.1.2 Frontend.....	6
2.1.3 R/3 System Administration	6
2.1.4 Functional Test of the conventional (insecure) environment	6
2.2 Installing the security product on the application server	7
2.3 Installing the security product on the frontend machines	7
2.4 GSSTEST on the application server	7
2.5 GSSTEST on the frontend machines	7
2.6 Enabling SNC on the R/3 application server	8
2.7 Enabling SNC on the frontend machines	9
2.8 SNC-Name retrieval via SAPlogon	9
2.8.1 Server Selection.....	10
2.8.2 Group Selection / "Load Balancing"	11
2.9 Secure R/3 access with SAPgui	12
2.10 Secure printing with SAPIpd	13
2.10.1 Configuring SAPIpd at the frontend PCs	13
2.10.2 Configuring the secure printer output device within R/3	15
2.11 Testing Secure RFC-connections, front-end to R/3 (SAPINFO)	18
2.11.1 RFC with specific application servers (RFC Type A)	18
2.11.2 RFC with load-balancing (RFC Type B)	19
2.12 Secure RFC-connections R/3 to RFC-Server programs (RFCEXEC)	20
2.12.1 Start at frontend.....	20
<hr/>	
3 Reading the output protocol of GSSTEST	22
3.1 GSSTEST output for the distributed GSSNTLM.DLL.....	22
3.2 More output for security products with message protection	40

1 Test Plan SNC (Secure Network Communications)

Secure Network Communications (SNC) is a layer in the SAP R/3 software to integrate and interface to third party security software that conforms to the *Generic Security Service API Version 2* (GSS-API v2) specification. This standard is being developed in the Internet Engineering Task Force (IETF), an international standardizations body. Through SNC, strong authentication, integrity protection and confidentiality services of external security products can be used by the distributed components of the SAP R/3 Software to protect their network communication.

To guarantee the interoperability of external security products with the SAP Software the external product has to be certified for the BC-SNC interface within SAP's Complementary Software Program (CSP). This document describes the certification tests that have to be passed by a security product to receive the BC-SNC certificate.

For general information about SNC functionality see the White Papers at <http://www.sap.com/systemmanagement> (→ Security).

For more information about the BC-SNC certification go to the Web page <http://www.sap.com/products/comsoft/scenarios/bc/bcover.htm>

1.1 Test Objective

The BC-SNC certification tests cover installation and configuration of the security product together with SAP Software client and server components. After installation and configuration, the test analyses the runtime behaviour of the third party security software, simulating the behaviour of SAP Software components that use the SNC layer. Additionally, the standalone GSSTEST test tool collects statistical data on the runtime performance of individual GSS-API function calls, on the characteristics and attributes of names, credentials and security contexts.

The return values of all API calls are checked for conformance to the GSS-API v2 Standard. Furthermore, the conformance to certain SAP-specific constraints on parameter values, token sizes and runtime behaviour is verified. Besides the functional test, GSSTEST simulates common configuration and usage errors caused by misspelled names for credential owners and for security context targets and logs the observed behaviour into the output protocol.

To further validate the interoperability with the SAP Software components secure connections and sessions have to be setup and used for the SAPgui (SAP Graphical User Interface), RFC (Remote Function Call) and remote printing with the SAPIpd.

1.2 Test Environment

The BC-SNC certification procedure requires the following preconfigured hardware and software.

Provided by SAP:

- A Microsoft Windows NT 4.0 Server running a 4.0B R/3 application server (pl>=417), connected to a Release 4.0B R/3-System.
- A Microsoft Windows NT 4.0 Server or Workstation with a local SAP Frontend installation for Win32, R/3 Release 4.0B or newer.
- A Microsoft Windows 95 PC with a local SAP Frontend installation for Win32, R/3 Release 4.0B or newer.

- The GSS-API test tool "GSSTEST" for BC-SNC certification, precompiled for the hardware platforms from above.

It is necessary that the hostnames of all these machines can be correctly resolved via DNS. When DHCP is used, this will require the use of DNS-Servers that will be dynamically updated from the DHCP server. If dynamic update of DNS with the DHCP information is not available or not configured, static IP addresses and static DNS entries will have to be used.

Provided by Partner:

- The third party security software to be examined for interoperability with SAP Software components through the BC-SNC interface supporting the hardware platforms from above.
- Any installation and configuration tools required as part of the security product's infrastructure.
- Preinstalled SAP R/3 Release 4.0B application server and system, as well as Frontend installation and security product installations on additional hardware other than the above (either bring to SAP certification site or on-site certification at Partner or Customer site).

1.3 Test Scenario Overview

The interoperability of the third party security software and SAP Software components over the BC-SNC-Interface are tested by the following steps and scenarios:

- Installation and configuration of the third party security software on the SAP R/3 application server.
- Installation and configuration of the third party security software on the SAP Frontend computer.
- Examination of shared library supplied by the third party security software with the standalone test tool GSSTEST on application server and frontend machines.
- Configuring/enabling the third party security software for the R/3 application server (R/3 and security product settings).
- Configuring/enabling the third party security software for the R/3 frontend components (SAPlogon, SAPgui, SAPlpd, rfcinfo, rfcexec).
- Testing SNC-Name retrieval via SAPlogon for starting SAPguis directly and with load balancing.
- Testing secure R/3 access with SAPgui.
- Testing secure printing with SAPlpd.
- Testing secure RFC connections in different scenarios using sapinfo and rfcexec.

2 The Test procedure

This chapter describes details of the test procedure; required steps and actions.

2.1 Preparing the environment

Before the certifications tests can be performed the test environment has to be setup as follows:

2.1.1 Application Server

Installation and configuration of an R/3 application server running R/3 Release 4.0B on Windows NT 4.0 against a 4.0B R/3 System and database with SNC disabled.

Dialog Workprocesses: >= 2

Spool Process: 1

2.1.2 Frontend

Installation and configuration of an R/3 4.0B Frontend (or newer) running on Windows NT 4.0 and Windows 95 with the components SAPlogon, SAPgui, SAPlpd and the RFC SDK.

Minimal Frontend installation using SAPsetup:

Select from "Individual Install" component list:

Desktop Development Kit,

SAPGUI 32-bit,

SAPlogon

2.1.3 R/3 System Administration

Use transaction **SU01** *User Maintenance* to create two R/3 User Accounts:

SNCTEST1 Logon data --> User type: Dialog

SNCTEST2 Logon data --> User type: Dialog

Use transaction **SMLG** *CCMS: Maintain Logon Groups* to define two logon groups:

PUBLIC containing application servers that permit insecure logon

SNC containing the R/3 application server that will be configured for SNC-logon

2.1.4 Functional Test of the conventional (insecure) environment

Start the R/3 application server. Watch the console output and trace files (if required) to verify the successful establishment of the connection to the database and the correct status of the work and spool processes.

Start SAPlogon and configure an entry to logon to the R/3 test system. Launch a SAPgui and logon to the R/3 test system with the SNCTEST1 and SNCTEST2 users using the conventional password-based mechanism.

2.2 Installing the security product on the application server

Complete any steps required to install the security product on the application server.

An R/3 application server requires initiating and accepting credentials referring both to the same identity/name for its operation. If either of both types of these credentials are short-lived (several hours or a few days), the procedure for automatic credentials refresh must be documented in the test report.

2.3 Installing the security product on the frontend machines

Complete any steps required to install the security product on the frontend machines running Windows NT 4.0 and Windows 95.

Most components of the SAP Frontend require only the default initiating credentials for operation and connecting to R/3-Systems. SAPIpd requires the availability of accepting credentials, and gsstest requires availability of both, initiating and accepting credentials, however, these are *not* required to refer to the same identity/name.

2.4 GSSTEST on the application server

Start the GSSTEST tool on the application server and provide the path to the security product library which contains the GSS API v2 functions and SNC adapter.

```
gsstest -l <drive>:\path\to\your\snclib.dll
        -a <target_name>
        -p appserv-nt.log
```

If the security product offers special configuration or usage options that will affect the operation of GSSTEST at the GSS-API level, these must be documented in the test report. Examples of such options are: use of software- or hardware-based authentication, possibility to configure different cryptographic algorithms for confidentiality and integrity for GSS_C_DEFAULT_QOP.

Save/Archive the output protocol "**appserv-nt.log**" created by the GSSTEST tool and interpret its contents according to Section 3. „Reading the Output Protocol of GSSTEST“.

2.5 GSSTEST on the frontend machines

Start the GSSTEST tool on the frontend machine and provide the path to the security product library which contains the GSS API v2 functions and SNC adapter.

```
gsstest -l <drive>:\path\to\your\snclib.dll
        -a <target_name>
        -p frontend-xx.log
```

If the security product offers special configuration or usage options that will affect the operation of GSSTEST at the GSS-API level, these must be documented in the test report. Examples of such options are: use of software- or hardware-based authentication, possibility to configure different cryptographic algorithms for confidentiality and integrity for GSS_C_DEFAULT_QOP.

Run GSSTEST additionally on both frontend platforms once without valid credentials. Although this will cause GSSTEST to abort prematurely, we need the resulting log file with the error messages that indicate the absence of valid credentials.

Save/Archive the output protocol "**frontend-xx.log**" created by the GSSTEST tool and interpret its contents according to Section 3. „Reading the Output Protocol of GSSTEST“.

2.7 Enabling SNC on the frontend machines

Set the environment variable SNC_LIB to contain the path to the security product library:

```
SNC_LIB = <drive>:\path\to\your\snclib.dll
```

Windows NT: Control Panel ⇒ System ⇒ Environment

Windows 95: edit AUTOEXEC.BAT and restart Windows 95

(Unix: add a definition of the environment variable SNC_LIB into the shell login script of the user which contains the full path and filename of the shared library of the security product. Some security products may require the definition of another environment variable to resolve non-standard shared library dependencies — refer to the documentation and requirements of the security product.

AIX:	LIB_PATH)
Digital UNIX, Solaris, Sinix:	LD_LIBRARY_PATH)
HP-UX:	SHLIB_PATH)

2.8 SNC-Name retrieval via SAPlogon

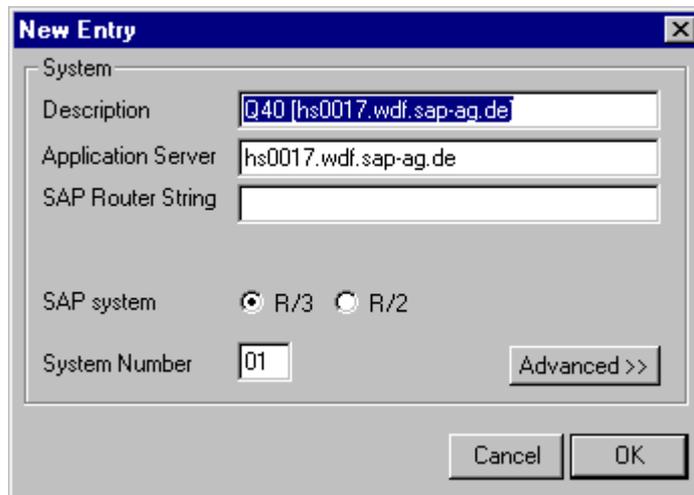
Start SAPlogon.



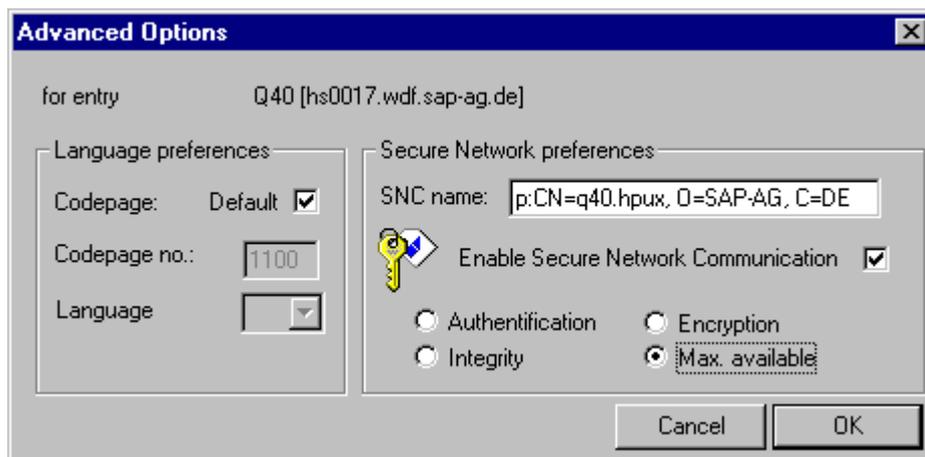
Create the two new entries as described in the next sections.

2.8.1 Server Selection

Add a new entry for the R/3 test system using "Server Selection".



In the 'New Entry' dialog, open the 'Advanced Options' and configure the SNC parameters.

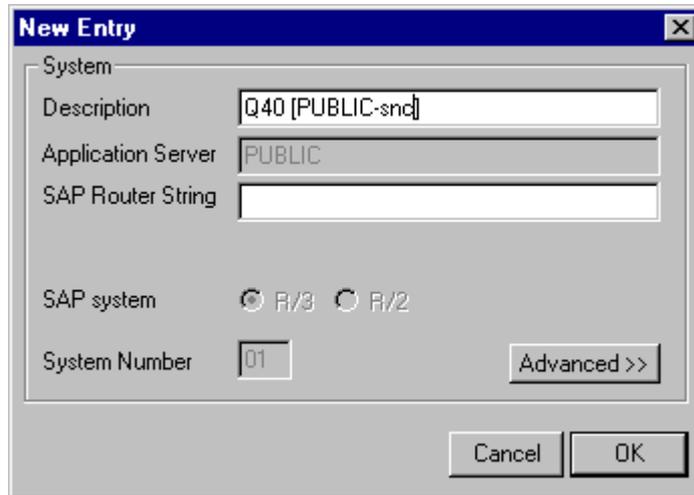


Since the SNC-Name of the application server is distributed by the message server, it should already appear in its field. Activate the checkbox to enable SNC and select 'Max. available' for the message protection (it is the default).

2.8.2 Group Selection / "Load Balancing"

Logon groups can be defined within R/3 using transaction **SMLG**.

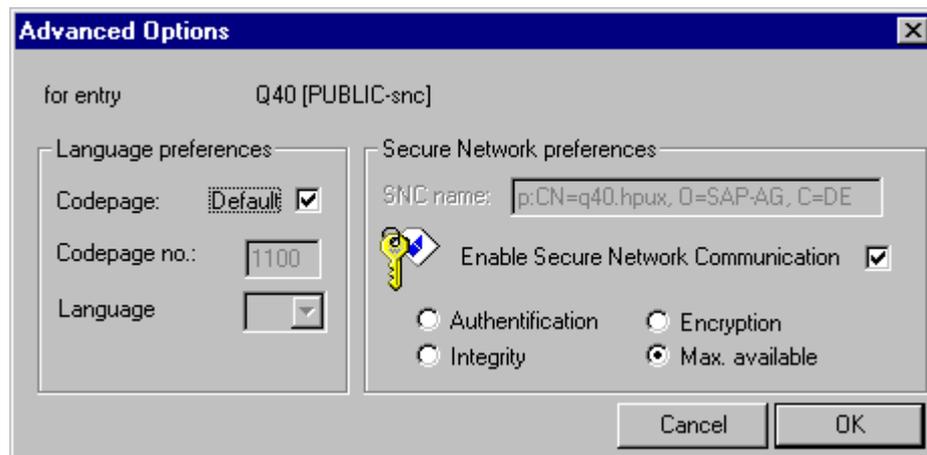
Add a new entry for the R/3 test system using "Group Selection".



The 'New Entry' dialog box is shown with the following fields and options:

- Description:** Q40 [PUBLIC-snc]
- Application Server:** PUBLIC
- SAP Router String:** (empty)
- SAP system:** R/3 R/2
- System Number:** 01
- Buttons:** Advanced >>, Cancel, OK

In the 'New Entry' dialog, open the 'Advanced Options' and configure the SNC parameters.



The 'Advanced Options' dialog box is shown for entry 'Q40 [PUBLIC-snc]' with the following settings:

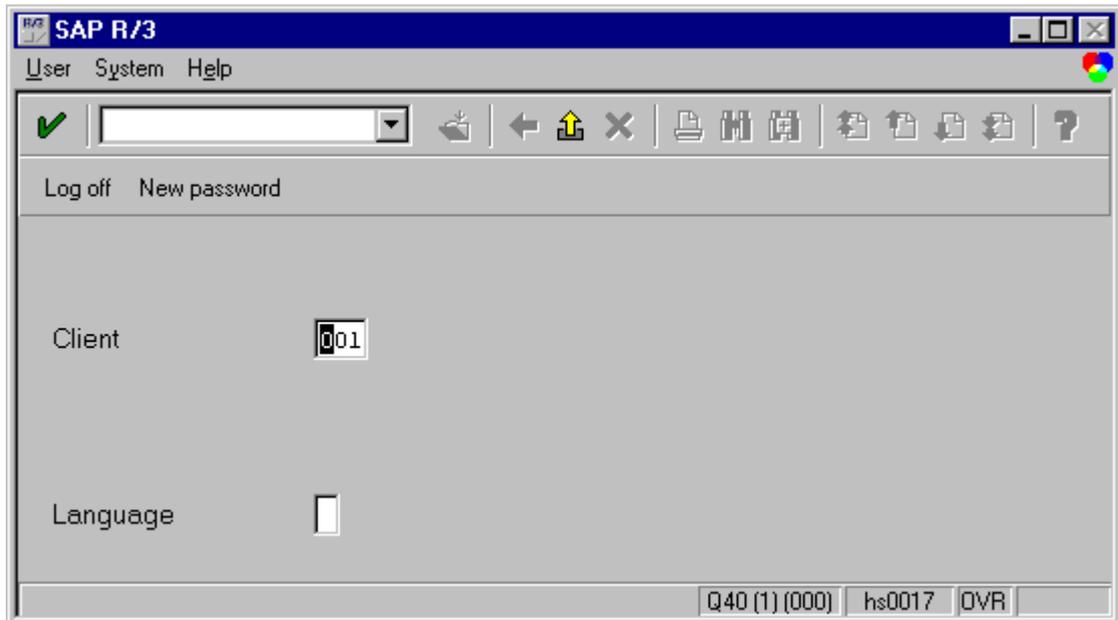
- Language preferences:**
 - Codepage:** Default
 - Codepage no.:** 1100
 - Language:** (dropdown menu)
- Secure Network preferences:**
 - SNC name:** p:CN=q40.hpux, O=SAP-AG, C=DE
 - Enable Secure Network Communication:**
 - Options:**
 - Authentication
 - Encryption
 - Integrity
 - Max. available
- Buttons:** Cancel, OK

When using Logon Groups instead of Server Selection, the actual application server name and its SNC-Name will be requested from the message server just before the SAPgui is started. Since the application server is dynamically selected from the message server each time the SAPlogon group item is activated, one can not preconfigure the SNC-Name. The field for the SNC-Name is greyed out and cannot be changed. Activate the checkbox to enable SNC and select 'Max. available' for the message protection (it is the default).

2.9 Secure R/3 access with SAPgui

First, verify that you have valid client credentials available for one of the accounts SNCTEST1 and SNCTEST2 that you have created in step 2.1.3 *R/3 System Administration* and assigned to externally authenticated SNC-Names in step 2.6 *Enabling SNC on the R/3 application server*.

Use the two previously configured entries from the list of SAPlogon to start SAPgui with SNC-protected communication. The initial logon screen should only show the fields **Client** and **Language**; the two other fields User and Password will not be shown when the user was externally authenticated.



You may change the client number if you want to logon to an account that is not in the default client, and you may specify a logon language. If you proceed from this screen with <Enter>, or the green checkmark button in the upper left, then the system will try to map your externally authenticated name to an R/3 account. If there is no user account matching your externally authenticated name in the selected client, then your logon request will be denied with an error message in the status bar at the bottom of the SAPgui window.

Logon to R/3 and open some extra windows using the menu path "System->Create Session"

Open the Status window on every new window using the menu path "System->Status...".

Close the first two windows identified by (1) and (2) in the window status bar, leftmost field, just behind the name of the R/3-System.

Open two more windows using the menu path "System->Create Session" and request the status window using menu path "System->Status..." again.

Start another SAPgui while the old session is still running and select "User->Copy Session" on the initial logon screen (shown above). The open windows from the previous session should all disappear and then the status windows should all reappear. The old SAPgui will terminate and exit when the new one takes over the existing session. (The SAPgui windows below the Status windows will probably not reappear until the Status windows are closed.)

2.10 Secure printing with SAPIpd

2.10.1 Configuring SAPIpd at the frontend PCs

Besides the procedure in Section 2.7: "Enabling SNC on the frontend machines", secure printing via SAPIpd requires additional configuration. SAPIpd operates as a standalone server and a security context acceptor at the GSS-API level, so it requires its own accepting credentials independent of the credentials of the user that may be working at the PC.

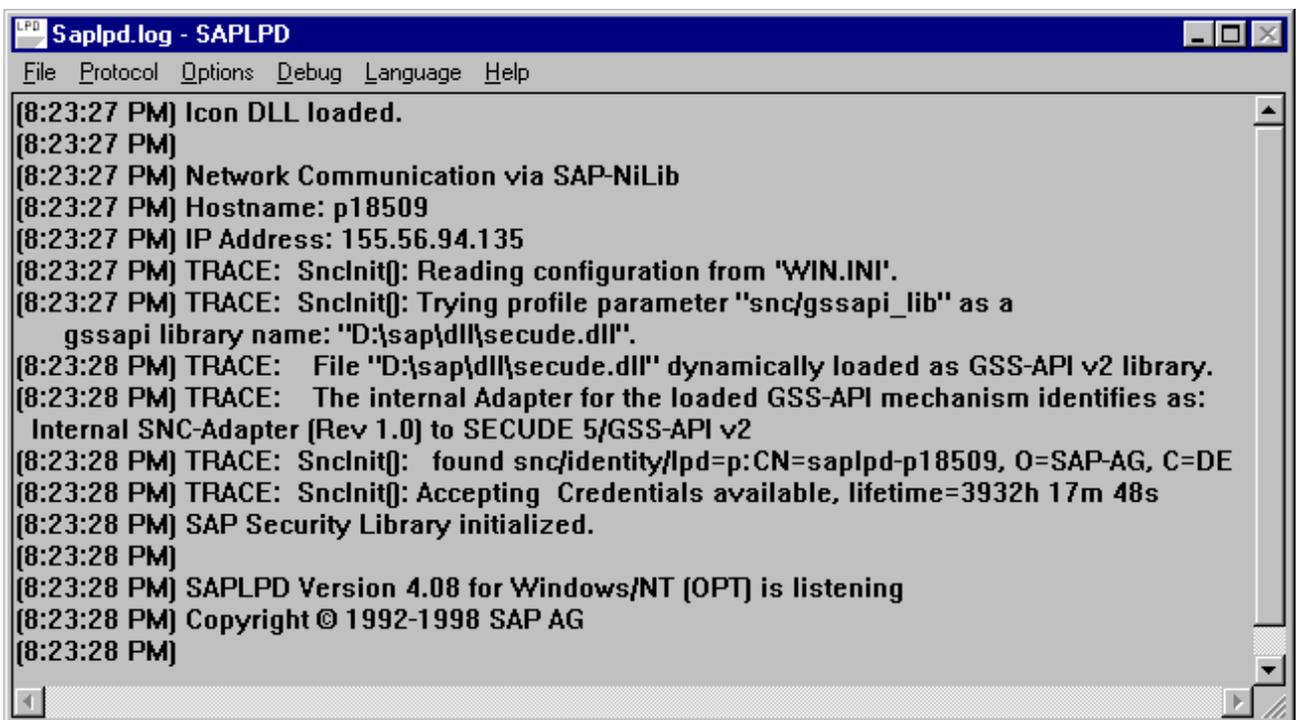
SAPIpd reads SNC-specific from the *.ini-File "SAPLPD.INI" that will be searched in the directory of the SAPLPD.EXE binary or in the Windows directory. If this file does not exist or does not contain a section [snc] with the entry "enable=1", SAPIpd will also search "WIN.INI" for SNC-specific information.

Sample SAPIpd configuration in SAPLPD.INI (or WIN.INI):

```
[ snc ]
enable=1
identity/lpd=<SNC-Name_of_saplpd>
gssapi_lib=<drive>:\path\to\your\snc\lib.dll
```

The line with "gssapi_lib=" can be omitted when the environment variable SNC_LIB is configured to be globally visible to all processes.

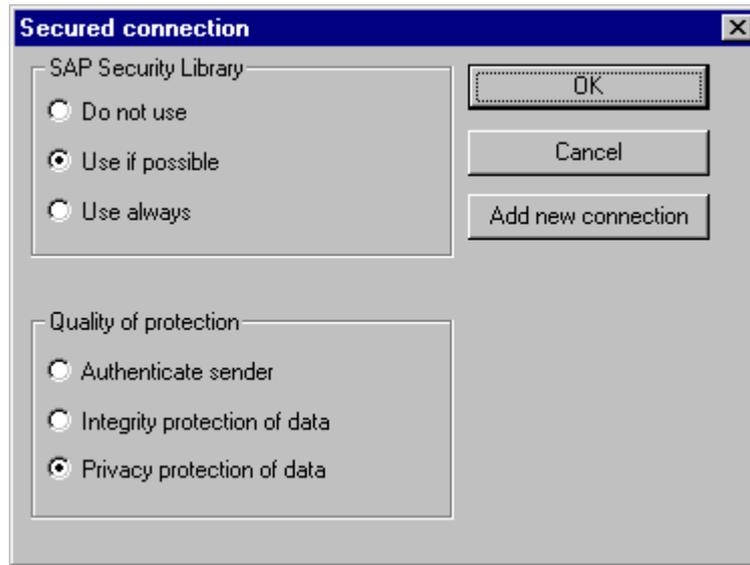
SAPIpd will always try to acquire its accepting credentials by specifying the explicit name from the configuration parameter "identity/lpd". The third-party security product may require additional configuration and setup changes to provide separate accepting credentials to SAPIpd that will not interfere with credentials of interactive users, who may be independently using secure SAPgui with this PC.



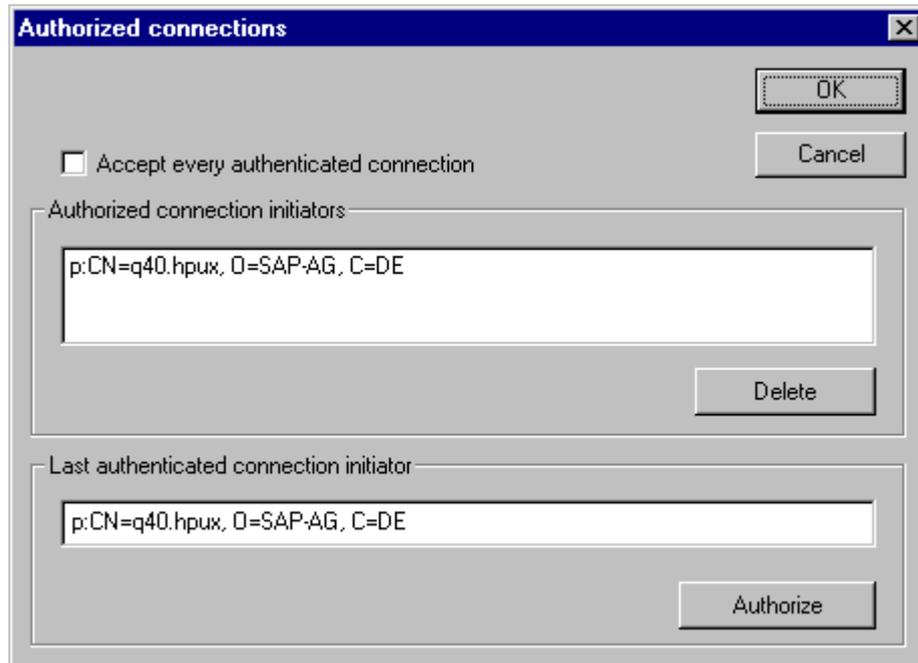
The screenshot shows a window titled "Saplpd.log - SAPLPD" with a menu bar containing "File", "Protocol", "Options", "Debug", "Language", and "Help". The log content is as follows:

```
(8:23:27 PM) Icon DLL loaded.
(8:23:27 PM)
(8:23:27 PM) Network Communication via SAP-NiLib
(8:23:27 PM) Hostname: p18509
(8:23:27 PM) IP Address: 155.56.94.135
(8:23:27 PM) TRACE: Snclnit(): Reading configuration from 'WIN.INI'.
(8:23:27 PM) TRACE: Snclnit(): Trying profile parameter "snc/gssapi_lib" as a
gssapi library name: "D:\sap\dll\secude.dll".
(8:23:28 PM) TRACE: File "D:\sap\dll\secude.dll" dynamically loaded as GSS-API v2 library.
(8:23:28 PM) TRACE: The internal Adapter for the loaded GSS-API mechanism identifies as:
Internal SNC-Adapter (Rev 1.0) to SECUDE 5/GSS-API v2
(8:23:28 PM) TRACE: Snclnit(): found snc/identity/lpd=p:CN=saplpd-p18509, O=SAP-AG, C=DE
(8:23:28 PM) TRACE: Snclnit(): Accepting Credentials available, lifetime=3932h 17m 48s
(8:23:28 PM) SAP Security Library initialized.
(8:23:28 PM)
(8:23:28 PM) SAPLPD Version 4.08 for Windows/NT (OPT) is listening
(8:23:28 PM) Copyright © 1992-1998 SAP AG
(8:23:28 PM)
```

Start SAPIpd, select "Options->Secured Connection" from the menu and the following dialog box will open:



Select **Use if possible** from the list for the SAP Security Library, select **Privacy protection of data** from the list for Quality of protection,



and press the button **Add new connection** to go ahead to the maintenance of the Access Control List (ACL) of SAPIpd.

Enter the SNC-name of the application server(s) that will be transferring print jobs securely to this SAPIpd into the field **Last authenticated connection initiator** and hit the button **Authorize** to add this name into the list of authorized connection initiators.

Close the dialog boxes again by hitting their OK buttons.

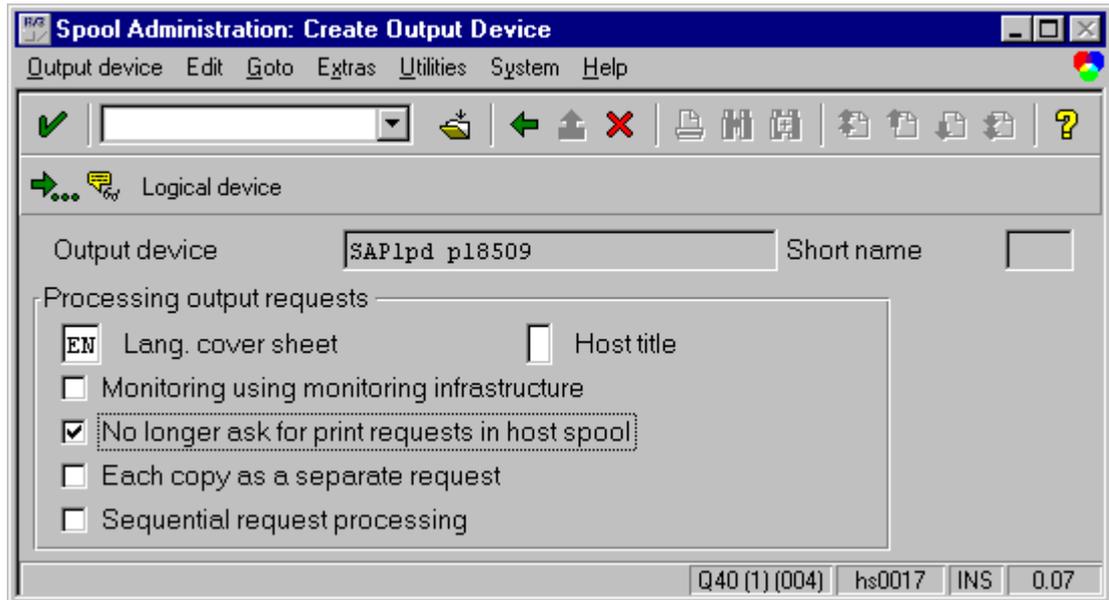
2.10.2 Configuring the secure printer output device within R/3

Create a new Printer with the transaction **SPAD** ("Spool Administration"). Press the Button for "Output Devices" and create a new output device (Printer):

Fill in the fields **Output device**, **Device type** and **Host printer**.

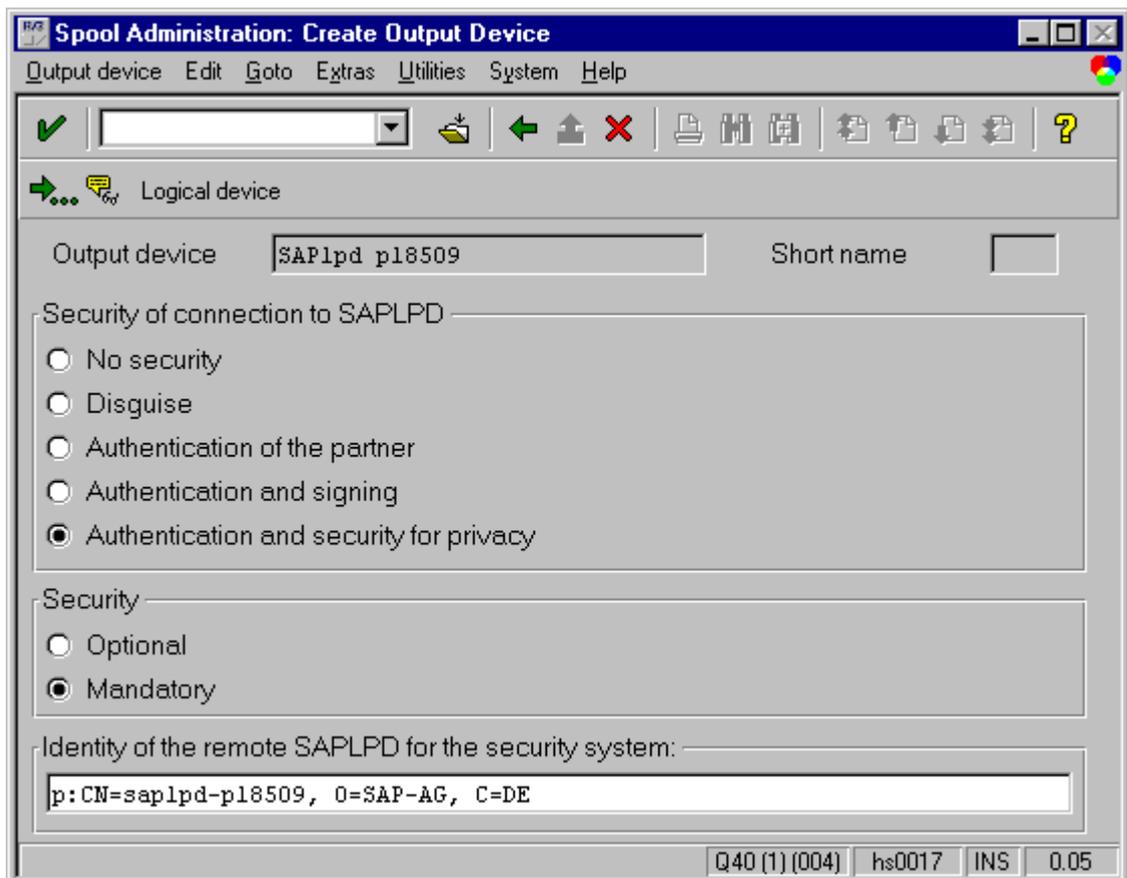
Enter a capital "S" for "SAPlpd" into the field **Access method to host spool** and hit <Return> on the keyboard: an additional field to enter the **Destination host** will appear on the screen. Enter the hostname of the frontend PC.

Go ahead to the next screen (<F5>).



Checkmark the option **No longer ask for print requests in host spool**.

Go ahead to the next screen (<F5>).



Select **Authentication and security for privacy** from the list of security characteristics for the SAP1pd connection.

Hit **<Enter>** on the keyboard: additional entry fields will appear.

Select **Mandatory** security.

Enter the SNC-Name of the credentials that SAPlpd on the frontend machine will be using for accepting the connections.

Save the printer definition.

Return to the list of output devices with the 'back' arrow.

Create a print job by printing the list of output devices using the Menu path "System->List->Print".

Select the **Output Device** that you just created with the F4-selection or just enter its name.

Checkmark the spool control option Print immed.

Submit the print request with the printer icon in the upper left.

You can track status and progress of the print request with transaction **SP01**. If an error occurs on SAPlpd for any reason, processing of output requests for this the output device will be suspended for several minutes. To reactivate the processing of this output device immediately, use transaction SPAD, select the output device from the list and open the definition for change. You can now reactivate the output device with <Ctrl-F2> or Menu "Edit->Reactivate".

2.11 Testing Secure RFC-connections, front-end to R/3 (SAPINFO)

Create an RFC configuration file SAPRFC.INI. This file should either be located in the same directory as the program executable, or the environment variable RFC_INI must be used to specify the full path and filename to this file.

The following test procedure is using SAPINFO.EXE, which is contained in the *Desktop Development Kit* of the frontend software. It lives in the directory

```
<drive>:\path\of\frontend\SAPGUI\RFCSDK\BIN\
```

To run this test, it may be preferable to copy the programs SAPINFO.EXE and RFCINFO.EXE, the file LIBRFC32.DLL into a separate/new directory and create the SAPRFC.INI file there.

2.11.1 RFC with specific application servers (RFC Type A)

An RFC-destination (Type A) in SAPRFC.INI is built with the following pattern:

```
DEST=<choose_your_name>
TYPE=A
ASHOST=<hostname_of_R/3_AppServer>
SYSNR=<system-nr_of_R/3>
```

When using SNC, one must specify additional parameters:

```
SNC_MODE=1
SNC_PARTNERNAME=<SNC-Name_of_R/3_AppServer>
SNC_LIB=<drive>:\path\to\your\snclib.dll
```

Here is a sample SAPRFC.INI with 3 destinations:

```
/*=====*/
/*  Type A:  R/3 system - specific application server      */
/*=====*/

/* Conventional access */
DEST=Q40_hs0017
TYPE=A
ASHOST=hs0017
SYSNR=01

/* Access using SNC-protected communication */
DEST=snc_Q40_hs0017
TYPE=A
ASHOST=hs0017
SYSNR=01
SNC_MODE=1
SNC_PARTNERNAME=p:CN=Q40.hpux, O=SAP-AG, C=DE
SNC_LIB=D:\sap\dll\secude.dll

/* Tickle an SNC-error by specifying a wrong target name */
DEST=error_Q40_hs0017
TYPE=A
ASHOST=hs0017
SYSNR=01
SNC_MODE=1
SNC_PARTNERNAME=p:CN=error, O=SAP-AG, C=DE
SNC_LIB=D:\sap\dll\secude.dll
```

Use this sample to create a SAPRFC.INI matching the R/3 test environment.

Now run the SAPINFO program with each of the 3 destinations:

1. SAPINFO dest=Q40_hs0017
2. SAPINFO dest=snc_Q40_hs0017
3. SAPINFO dest=error_Q40_hs0017

The first two destinations should succeed, the third destination should fail with an SNC-Error (provided that the security product provides mutual authentication at the GSS-API level). The third test/destination is primarily to check whether SNC is actually being used and working.

2.11.2 RFC with load-balancing (RFC Type B)

Load-balancing or "group-logon" dynamically retrieves the target SNC-Name from the message server. Logon groups can be defined within R/3 using transaction **SMLG**.

An RFC-destination (Type B) in SAPRFC.INI is built with the following pattern:

```
DEST=<choose_your_name>
TYPE=B
R3NAME=<sid>
MSHOST=<hostname_of_R/3_MessageServer>
GROUP=<name_of_SNC_logon_group>
```

When using SNC, one must specify additional parameters:

```
SNC_MODE=1
SNC_PARTNERNAME=p:unused
SNC_LIB=<drive>:\path\to\your\snc\lib.dll
```

Here is a sample SAPRFC.INI with 2 destinations:

```
/*=====*/
/* Type B: R/3 system - load balancing feature */
/*=====*/

/* Conventional access */
DEST=Q40_PUBLIC
TYPE=B
R3NAME=Q40
MSHOST=hs0017
GROUP=public

/* Access using SNC-protected communication */
DEST=Q40_SNC
TYPE=B
R3NAME=Q40
MSHOST=hs0017
GROUP=SNC
SNC_MODE=1
SNC_PARTNERNAME=p:unused
SNC_LIB=D:\sap\dll\secude.dll
```

Use this sample to create a SAPRFC.INI matching the R/3 test environment.

Now run the SAPINFO program with each of the 2 destinations:

1. SAPINFO dest=Q40_PUBLIC
2. SAPINFO dest=Q40_SNC

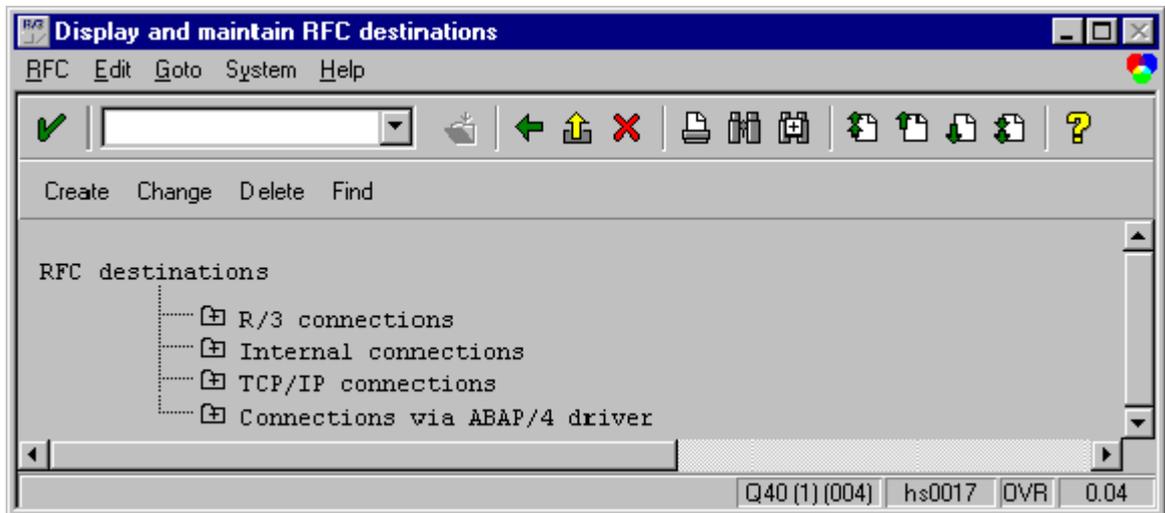
Traces of SNC-related activities can be enabled by setting the environment variable CPIC_TRACE=2. A trace file starting with the letters "CPIC" will be written into the current directory for every new process/program that is started.

2.12 Secure RFC-connections R/3 to RFC-Server programs (RFCEXEC)

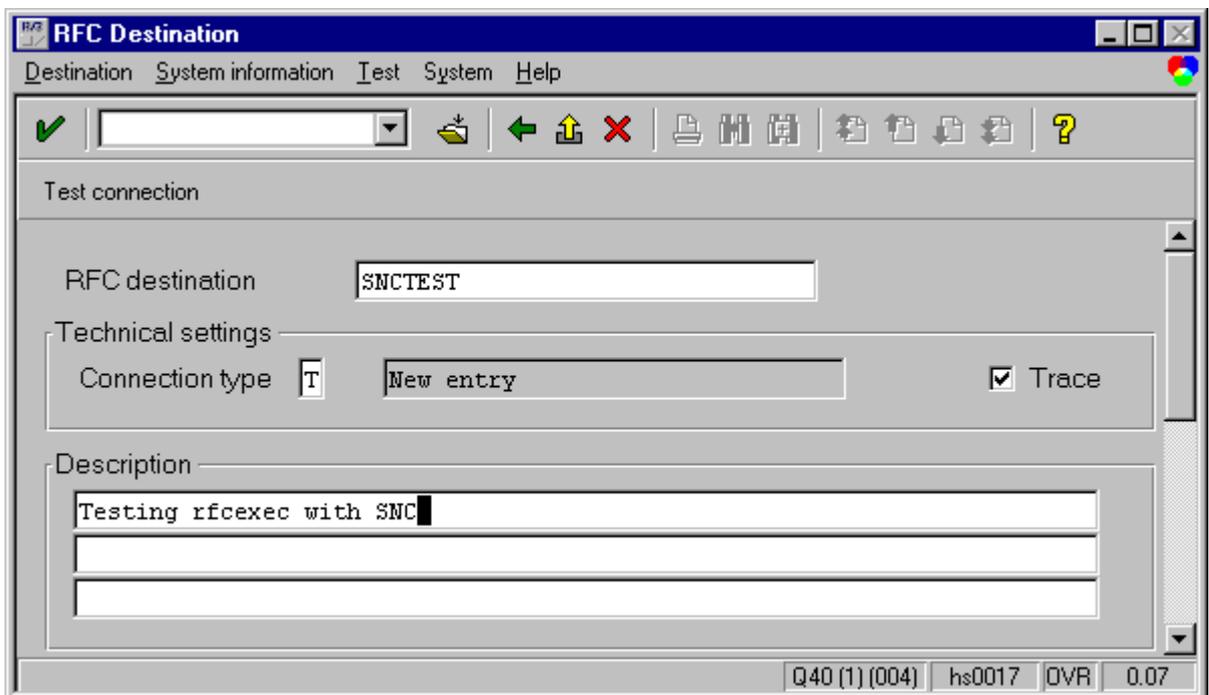
2.12.1 Start at frontend

Log on to the R/3-System using a secure frontend.

Start transaction **SM59** to create a new RFC destination:



Hit the **Create** button.



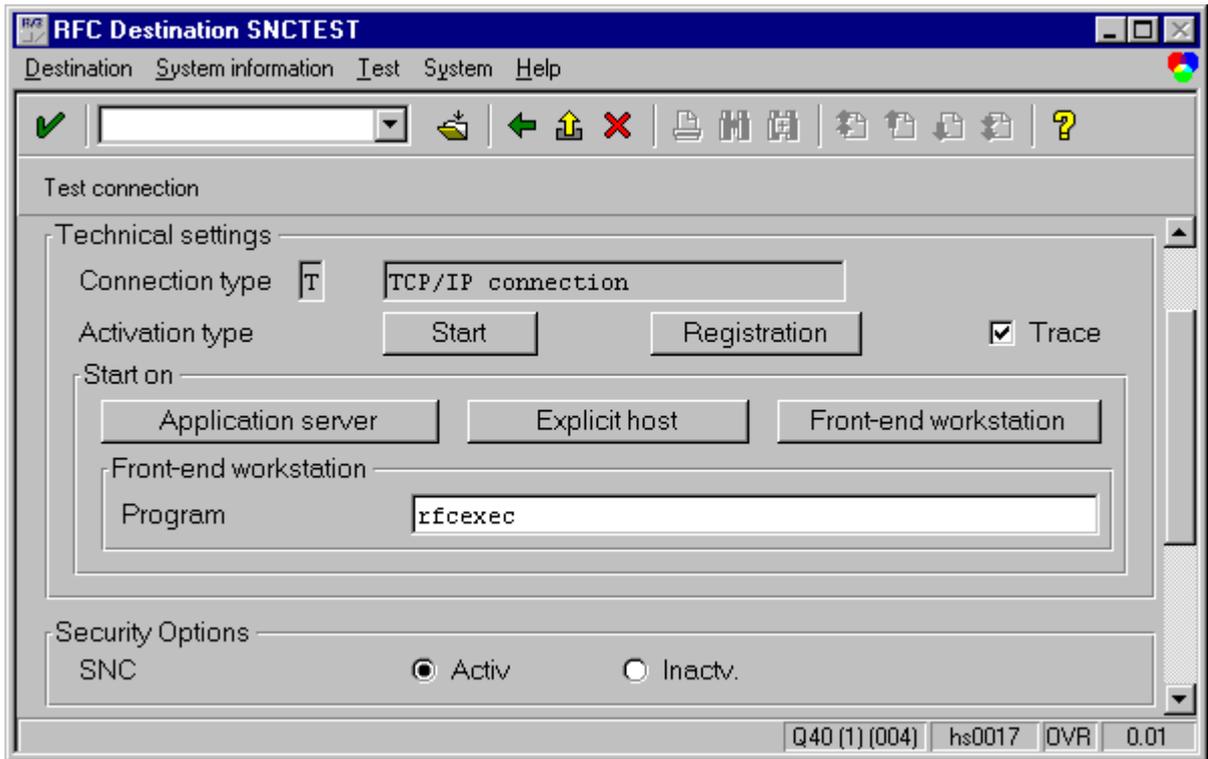
Enter the name of the RFC destination "SNCTEST".

Enter the connection type "T" for "TCP/IP connection".

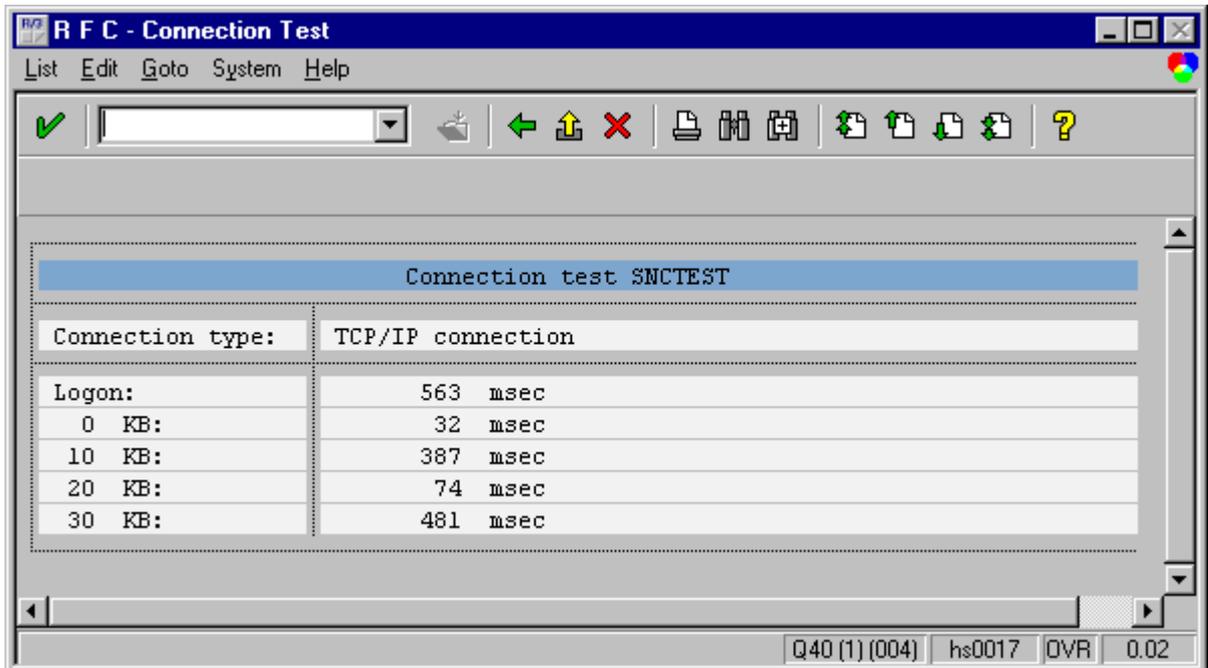
Checkmark the **Trace** option.

Enter a description into the description field.

Hit <Enter> on the keyboard.



- Select the *Activation type* button **Start**.
- Select the *Start on* location button **Front-end workstation**.
- Enter **rfcexec** as the name of the Program.
- Select *Security Option* SNC as **Active**.
- Save the RFC destination with the save icon in the toolbar.
- Hit the button **Test connection**.



3 Reading the output protocol of GSSTEST

When the test of the security product completes, GSSTEST compiles an output summary on the test results. At the end of that summary, a final rating is printed whether the security product met the technical interoperability requirements for the SNC Interface in SAP R/3 during the test. It will look similar to this one:

```
=====
Passing all API result tests.
Passing all SAP constraints.

--- Passed ---   SAP GSS-API v2 Test (builtin SNC-Adapter)

Mechanism   = {1 3 6 1 4 1 694 2 1 2}           MECH= SAPntlm SSO (NT4/Win95)
Nametype    = {1 3 6 1 4 1 694 2 1 2 1}         NT= GSS_SAP_NT_DOMAIN_USER

Max. data protection level =   1 (Authentication only)

Hardware Platform           =   Microsoft Windows NT 4.0 (Build 1381)

=====
```

Besides the functionality of the security product according to the GSS-API specification, GSSTEST also verifies certain interoperability requirements with SNC / SAP R/3, indicated as *SAP constraints*.

For the BC-SNC certification, all API and SAP constraints tests must be passed.

3.1 GSSTEST output for the distributed GSSNTLM.DLL

The following is the sample output of GSSTEST for GSSNTLM.DLL, a GSS-API v2 wrapper for the NTLM SSP in Microsoft Windows NT and 95/98:

```
*****
***                                     ***
*** "gsstest" -- GSS-API v2 Shared Library API Test Program                 ***
***                                     ***
*** Version 1.05      27-Nov-1998                                           ***
***                                     ***
*** This implementation is Copyright (c), 1998 SAP AG Walldorf              ***
***                                     ***
*** This tool may be freely used to test functionality and                  ***
*** robustness of GSS-API v2 mechanism implemenations                      ***
*****
*** SAP AG DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,            ***
*** INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND                 ***
*** FITNESS. IN NO EVENT SHALL SAP AG BE LIABLE FOR ANY SPECIAL,           ***
*** INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER            ***
*** RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN           ***
*** ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,              ***
*** ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE           ***
*** OF THIS SOFTWARE.                                                       ***
*****
```

```
Timer resolution of QueryPerformanceCounter() is (at least)      0.009 millisec
1 second passed in      999.521 millisec.
```

Loading GSS-API DLL "gssntlm.dll" ...

```
mech_list from gss_indicate_mechs() contains 1 gss_OID element:
{
  [ 0] = {1 3 6 1 4 1 694 2 1 2}      MECH= SAPntlm SSO (NT4/Win95)
}
```

```
Selecting mechanism (0):
  {1 3 6 1 4 1 694 2 1 2}      MECH= SAPntlm SSO (NT4/Win95)
```

```
Checking supported nametypes via gss_inquire_names_for_mech()
name_types contains 3 gss_OID elements:
{
  [ 0] = {1 3 6 1 5 6 4}           NT= GSS_C_NT_EXPORTED_NAME
  [ 1] = {1 2 840 113554 1 2 1 1}  NT= GSS_C_NT_USER_NAME
  [ 2] = {1 3 6 1 4 1 694 2 1 2 1} NT= GSS_SAP_NT_DOMAIN_USER
}
```

=====
Testing generic gssapi functions ...

```
TEST: passing mech_list from indicate_mechs() to release_oid_set()
RESULT OK
TEST: passing name_types from inquire_names_for_mech() to release_oid_set()
RESULT OK
```

=====
Testing credentials management functions ...

```
TEST: *default* initiating credentials (acquire_cred default mechs)
RESULT OK
  actual_mechs from gss_acquire_cred() contains 1 gss_OID element:
  {
    [ 0] = {1 3 6 1 4 1 694 2 1 2}      MECH= SAPntlm SSO (NT4/Win95)
  }
```

```
-----
TEST: *default* initiating credentials (acquire_cred specific mechs)
RESULT OK
TEST: *default* initiating credentials (inquire_cred only)
RESULT OK
TEST: named default initiating credentials (acquire_cred with name)
RESULT OK
TEST: acquire_cred and inquire_cred with NO optional parameters
RESULT OK
  My own name/identity (from default creds) resolves to
  "SAP_ALL\D019080"
  Nametype oid = {1 3 6 1 4 1 694 2 1 2 1}      NT= GSS_SAP_NT_DOMAIN_USER
```

```
Framing details for exported name (Section 3.2, GSS-API v2 spec):
TOK_ID          : 00000: 04 01
MECH_OID_LEN = 12 : 00002: 00 0c
  OID tag       : 00004: 06
  OID len = 10  : 00005: 0a
  OID elements  : 00006: 2b 06 01 04 01 85 36 02 01 02
    = {1 3 6 1 4 1 694 2 1 2}      MECH= SAPntlm SSO (NT4/Win95)
NAME_LEN       = 15 : 00010: 00 00 00 0f
NAME           : 00014: 53 41 50 5f 41 4c 4c 5c      SAP_ALL\
                0001c: 44 30 31 39 30 38 30      D019080
```

Since you didn't give me a target name, I'll try to talk to myself!

```
TEST: acquiring *default* initiating credentials (simple)
RESULT OK
TEST: acquiring *default* initiating credentials (query)
```

```

RESULT OK
TEST: acquiring initiating credentials (gss_name_t)
RESULT OK
TEST: acquiring initiating credentials (printable name)
RESULT OK
TEST: acquiring initiating credentials (can. printable name)
RESULT OK
TEST: acquiring accepting credentials for target (printable name)
RESULT OK
TEST: acquiring accepting credentials for target (can. printable name)
RESULT OK
TEST: acquiring *default* accepting credentials (simple)
RESULT OK
TEST: acquiring *default* accepting credentials (query)
RESULT OK
=====
Testing names management functions ...
-----
TEST: Testing consistency of gss_name_t conversions
RESULT OK
TEST: Testing consistency of gss_name_t conversions
RESULT OK
=====
Context establishment functions ...
-----
TEST: Testing sec_context est.: ini_cred=SIMPLE, acc_cred=GSSNAMED
RESULT OK
TEST: Testing sec_context est.: ini_cred=CHECKED, acc_cred=GSSNAMED
RESULT OK
TEST: Testing sec_context est.: ini_cred=GSSNAMED, acc_cred=GSSNAMED
RESULT OK
TEST: Testing sec_context est.: ini_cred=PRNAMED, acc_cred=GSSNAMED
RESULT OK
TEST: Testing sec_context est.: ini_cred=PRNAMED_VIA_XP, acc_cred=GSSNAMED
RESULT OK
TEST: Testing sec_context est.: ini_cred=SIMPLE, acc_cred=CHECKED
RESULT OK
TEST: Testing 10 sec_context est.: ini_cred=CHECKED, acc_cred=GSSNAMED
RESULT OK
=====
Message Protection functions ...
-----
skipping -- GSS-API mechanism doesn't seem to support message protection.
=====
Testing gss_display_status() major_status translations ...
-----
TEST: Testing gss_display_status() for routine errors:
GSS_S_COMPLETE                               =>
_____ "All is well that ends well"
GSS_S_BAD_MECH                               =>
_____ "Unsupported mechanism"
GSS_S_BAD_NAME                               =>
_____ "Invalid name"
GSS_S_BAD_NAME_TYPE                          =>
_____ "Unsupported nametype"
GSS_S_BAD_BINDINGS                           =>
_____ "Bad channel bindings"
GSS_S_BAD_STATUS                             =>
_____ "Invalid status code"
GSS_S_BAD_MIC                                =>
_____ "MIC did not verify"
GSS_S_NO_CRED                                =>
_____ "No valid credentials provided"
GSS_S_NO_CONTEXT                             =>

```

	"No valid context specified"
GSS_S_DEFECTIVE_TOKEN	=>
	"Defective token provided"
GSS_S_DEFECTIVE_CREDENTIAL	=>
	"Defective credentials detected"
GSS_S_CREDENTIALS_EXPIRED	=>
	"Credentials have expired"
GSS_S_CONTEXT_EXPIRED	=>
	"Security context has expired"
GSS_S_FAILURE	=>
	"Miscellaneous Failure"
GSS_S_BAD_QOP	=>
	"Invalid QOP value supplied"
GSS_S_UNAUTHORIZED	=>
	"Operation unauthorized"
GSS_S_UNAVAILABLE	=>
	"Operation/Feature unavailable"
GSS_S_DUPLICATE_ELEMENT	=>
	"Credential element already exists"
GSS_S_NAME_NOT_MN	=>
	"Provided name is not a mechanism name"

RESULT OK

TEST: Testing gss_display_status() for informational status:

GSS_S_CONTINUE_NEEDED	=>
	"Further token exchanges required"
GSS_S_DUPLICATE_TOKEN	=>
	"Duplicate token detected"
GSS_S_OLD_TOKEN	=>
	"Old token detected"
GSS_S_UNSEQ_TOKEN	=>
	"Tokens not in sequential order"
GSS_S_GAP_TOKEN	=>
	"An earlier token is still missing"

RESULT OK

TEST: Testing gss_display_status() for calling errors:

GSS_S_CALL_INACCESSIBLE_READ	=>
	"Invalid pointer for input arg"
GSS_S_CALL_INACCESSIBLE_WRITE	=>
	"Invalid pointer for output arg"
GSS_S_CALL_BAD_STRUCTURE	=>
	"Invalid structure supplied"

RESULT OK

TEST: Testing gss_display_status() for status value combinations:

(GSS_S_FAILURE GSS_S_DUPLICATE_TOKEN)	=>
	"Miscellaneous Failure"
	"Duplicate token detected"
(GSS_S_FAILURE GSS_S_OLD_TOKEN)	=>
	"Miscellaneous Failure"
	"Old token detected"
(GSS_S_FAILURE GSS_S_UNSEQ_TOKEN)	=>
	"Miscellaneous Failure"
	"Tokens not in sequential order"
(GSS_S_FAILURE GSS_S_GAP_TOKEN)	=>
	"Miscellaneous Failure"
	"An earlier token is still missing"
(GSS_S_CALL_INACCESSIBLE_READ GSS_S_BAD_NAME)	=>
	"Invalid pointer for input arg"
	"Invalid name"
(GSS_S_CALL_INACCESSIBLE_WRITE GSS_S_DEFECTIVE_CREDENTIAL)	=>
	"Invalid pointer for output arg"
	"Defective credentials detected"
(GSS_S_CALL_BAD_STRUCTURE GSS_S_BAD_BINDINGS)	=>
	"Invalid structure supplied"
	"Bad channel bindings"

```

RESULT OK
TEST: Testing gss_display_status() for mean/invalid value combinations:
      (GSS_S_CALL_INACCESSIBLE_READ|GSS_S_DEFECTIVE_TOKEN|GSS_S_DUPLICATE_TOKEN) =>
          "Invalid pointer for input arg"
          "Defective token provided"
          "Duplicate token detected"

      (GSS_S_CALL_INACCESSIBLE_READ|GSS_S_DEFECTIVE_TOKEN|GSS_S_DUPLICATE_TOKEN|GSS_S_
CONTINUE_NEEDED) =>
          "Invalid pointer for input arg"
          "Defective token provided"
          "Further token exchanges required"
          "Duplicate token detected"
      (GSS_S_CONTINUE_NEEDED|GSS_S_DUPLICATE_TOKEN|GSS_S_UNSEQ_TOKEN) =>
          "Further token exchanges required"
          "Duplicate token detected"
          "Tokens not in sequential order"
      (0xffffffffful) =>
          "(255): Unknown calling error number"
          "(255): Unknown routine error number"
          "Further token exchanges required"
          "Duplicate token detected"
          "Old token detected"
          "Tokens not in sequential order"
          "An earlier token is still missing"
          "(0xffe0): Unknown informatory status"
      (0x04130020ul) =>
          "(4): Unknown calling error number"
          "(19): Unknown routine error number"
          "(0x0020): Unknown informatory status"
      (0x10200800ul) =>
          "(16): Unknown calling error number"
          "(32): Unknown routine error number"
          "(0x0800): Unknown informatory status"

RESULT OK
=====
Testing error handling of name management functions ...
-----
TEST: Feeding gss_import_name() with a mech_oid nametype
Status: gss_import_name() == (GSS_S_BAD_NAME_TYPE)
      gss_display_status(0x00030000,GSS_S_GSS_CODE) =
          "Unsupported nametype"

RESULT OK
TEST: Feeding gss_import_name() with garbage: "XXXXXXXX"
      parses to = "SAP_ALL\XXXXXXXX"

RESULT OK
TEST: Feeding gss_import_name() with garbage: "CBA/ZYX"
      parses to = "CBA\ZYX"

RESULT OK
TEST: Feeding gss_import_name() with garbage: "CBA\ZYX"
      parses to = "CBA\ZYX"

RESULT OK
TEST: Feeding gss_import_name() with garbage: "CBA@ZYX"
      parses to = "SAP_ALL\CBA@ZYX"

RESULT OK
TEST: Feeding gss_import_name() with garbage: "XY Z"
      parses to = "SAP_ALL\XY Z"

RESULT OK
TEST: Feeding gss_import_name() with garbage: "AB C"
      parses to = "SAP_ALL\AB C"

RESULT OK
TEST: Feeding gss_import_name() with garbage: "A B C D"
      parses to = "SAP_ALL\A B C D"

RESULT OK

```

```

TEST: Feeding gss_import_name() with garbage: "A@B@C@X"
      parses to = "SAP_ALL\A@B@C@X"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "A@@@@@Z"
      parses to = "SAP_ALL\A@@@@@Z"
RESULT OK
TEST: Feeding gss_import_name() with garbage:
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
      parses to =
"SAP_ALL\XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "a,b,c,d"
      parses to = "SAP_ALL\A,B,C,D"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "A/B/C/D"
Status: gss_canonicalize_name() == (GSS_S_BAD_NAME)
      gss_display_status(0x00020000,GSS_S_GSS_CODE) =
      "Invalid name"
      gss_display_status(0x10000003,GSS_S_MECH_CODE) =
      "Invalid characters in name"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "D\C\B\A"
Status: gss_canonicalize_name() == (GSS_S_BAD_NAME)
      gss_display_status(0x00020000,GSS_S_GSS_CODE) =
      "Invalid name"
      gss_display_status(0x10000003,GSS_S_MECH_CODE) =
      "Invalid characters in name"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "CN=A1,OU=B ,O = C,      C      =ES"
      parses to = "SAP_ALL\CN=A1,OU=B ,O = C,      C      =ES"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "CN=A2, OU=C, O=D, C=ES"
      parses to = "SAP_ALL\CN=A2, OU=C, O=D, C=ES"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "C=ES, O=C,      OU = B,CN=A3"
      parses to = "SAP_ALL\C=ES, O=C,      OU = B,CN=A3"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "OU=H, C=ES, CN=A4, O=J"
      parses to = "SAP_ALL\OU=H, C=ES, CN=A4, O=J"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "CN=A5; OU=Y; O=X; C=ES"
      parses to = "SAP_ALL\CN=A5; OU=Y; O=X; C=ES"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "CN = A6 , OU=T;O =U, C= ES"
      parses to = "SAP_ALL\CN = A6 , OU=T;O =U, C= ES"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "CN=A7,, OU=Z,O=Y,, ,C=ES"
      parses to = "SAP_ALL\CN=A7,, OU=Z,O=Y,, ,C=ES"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "CN=A8;; OU=T;O=S;; ;C=ES"
      parses to = "SAP_ALL\CN=A8;; OU=T;O=S;; ;C=ES"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "/CN=A8/OU=D/O=E/C=ES"
Status: gss_canonicalize_name() == (GSS_S_BAD_NAME)
      gss_display_status(0x00020000,GSS_S_GSS_CODE) =
      "Invalid name"
      gss_display_status(0x10000003,GSS_S_MECH_CODE) =
      "Invalid characters in name"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "/C=ES/O=F/OU=E/CN=A9"
Status: gss_canonicalize_name() == (GSS_S_BAD_NAME)
      gss_display_status(0x00020000,GSS_S_GSS_CODE) =
      "Invalid name"
      gss_display_status(0x10000003,GSS_S_MECH_CODE) =

```

```

    "Invalid characters in name"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "/OU=E/C=ES/CN=A10/O=G"
Status: gss_canonicalize_name() == (GSS_S_BAD_NAME)
        gss_display_status(0x00020000,GSS_S_GSS_CODE) =
            "Invalid name"
        gss_display_status(0x10000003,GSS_S_MECH_CODE) =
            "Invalid characters in name"
RESULT OK
TEST: Feeding gss_import_name() with garbage: "A:B%C/D\E+F-G@H"
Status: gss_import_name() == (GSS_S_BAD_NAME)
        gss_display_status(0x00020000,GSS_S_GSS_CODE) =
            "Invalid name"
        gss_display_status(0x10000003,GSS_S_MECH_CODE) =
            "Invalid characters in name"
        parses to = "A:B%C\D\E+F-G@H"
RESULT NOT ok (rc=1)
-----
TEST: Feeding gss_import_name() with garbage: "A@B:C%D/E\F+G-H"
Status: gss_import_name() == (GSS_S_BAD_NAME)
        gss_display_status(0x00020000,GSS_S_GSS_CODE) =
            "Invalid name"
        gss_display_status(0x10000003,GSS_S_MECH_CODE) =
            "Invalid characters in name"
        parses to = "A@B:C%D\E\F+G-H"
RESULT NOT ok (rc=1)
-----
TEST: Feeding gss_import_name() with garbage: ""
Status: gss_import_name() == (GSS_S_BAD_NAME)
        gss_display_status(0x00020000,GSS_S_GSS_CODE) =
            "Invalid name"
        gss_display_status(0x00000003,GSS_S_MECH_CODE) =
            "Invalid gss_buffer handle"
RESULT OK
=====
Testing error handling of credential management functions ...
-----
TEST: Feeding gss_acquire_cred(INITIATE) with a single nametype-OID mech_set
Status: gss_acquire_cred Ini() == (GSS_S_BAD_MECH)
        gss_display_status(0x00010000,GSS_S_GSS_CODE) =
            "Unsupported mechanism"
        gss_display_status(0x0000001e,GSS_S_MECH_CODE) =
            "Unknown mechanism requested"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with a single nametype-OID mech_set
Status: gss_acquire_cred Acc() == (GSS_S_BAD_MECH)
        gss_display_status(0x00010000,GSS_S_GSS_CODE) =
            "Unsupported mechanism"
        gss_display_status(0x0000001e,GSS_S_MECH_CODE) =
            "Unknown mechanism requested"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "XXXXXXXX"
    Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "XXXXXXXX"
        cred owner resolves to = "SAP_ALL\XXXXXXXX"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "CBA/ZYX"
    Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "CBA/ZYX"
        cred owner resolves to = "CBA\ZYX"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "CBA\ZYX"
    Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "CBA\ZYX"
        cred owner resolves to = "CBA\ZYX"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "CBA@ZYX"

```

```

Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "CBA@ZYX"
cred owner resolves to = "SAP_ALL\CBA@ZYX"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "XYZ"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "XY Z"
cred owner resolves to = "SAP_ALL\XY Z"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "AB C"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "AB C"
cred owner resolves to = "SAP_ALL\AB C"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "A B C D"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "A B C D"
cred owner resolves to = "SAP_ALL\A B C D"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "A@B@C@X"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "A@B@C@X"
cred owner resolves to = "SAP_ALL\A@B@C@X"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "A@@@@@Z"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "A@@@@@Z"
cred owner resolves to = "SAP_ALL\A@@@@@Z"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
cred owner resolves to =
"SAP_ALL\XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "a,b,c,d"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "a,b,c,d"
cred owner resolves to = "SAP_ALL\A,B,C,D"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "CN=A1,OU=B ,O = C,
C =ES"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "CN=A1,OU=B ,O =
C, C =ES"
cred owner resolves to = "SAP_ALL\CN=A1,OU=B ,O = C, C =ES"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "CN=A2, OU=C, O=D,
C=ES"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "CN=A2, OU=C, O=D,
C=ES"
cred owner resolves to = "SAP_ALL\CN=A2, OU=C, O=D, C=ES"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "C=ES, O=C, OU =
B,CN=A3"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "C=ES, O=C, OU =
B,CN=A3"
cred owner resolves to = "SAP_ALL\C=ES, O=C, OU = B,CN=A3"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "OU=H, C=ES, CN=A4,
O=J"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "OU=H, C=ES, CN=A4,
O=J"
cred owner resolves to = "SAP_ALL\OU=H, C=ES, CN=A4, O=J"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "CN=A5; OU=Y; O=X;
C=ES"
Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "CN=A5; OU=Y; O=X;
C=ES"
cred owner resolves to = "SAP_ALL\CN=A5; OU=Y; O=X; C=ES"
RESULT OK

```

```

TEST: Feeding gss_acquire_cred(INITIATE) with owner name "CN = A6 , OU=T;O =U,
C= ES"
  Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "CN = A6 , OU=T;O
=U, C= ES"
  cred owner resolves to = "SAP_ALL\CN = A6 , OU=T;O =U, C= ES"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "CN=A7,, OU=Z,O=Y,,
,C=ES"
  Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "CN=A7,, OU=Z,O=Y,,
,C=ES"
  cred owner resolves to = "SAP_ALL\CN=A7,, OU=Z,O=Y,, ,C=ES"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "CN=A8;; OU=T;O=S;;
;C=ES"
  Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "CN=A8;; OU=T;O=S;;
;C=ES"
  cred owner resolves to = "SAP_ALL\CN=A8;; OU=T;O=S;; ;C=ES"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "A:B%C/D\E+F-G@H"
  Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "A:B%C/D\E+F-G@H"
  cred owner resolves to = "A:B%C\D\E+F-G@H"
RESULT OK
TEST: Feeding gss_acquire_cred(INITIATE) with owner name "A@B:C%D/E\F+G-H"
  Hmmm ... gss_acquire_cred(INITIATE) happily accepted name "A@B:C%D/E\F+G-H"
  cred owner resolves to = "A@B:C%D\E\F+G-H"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "XXXXXXXXXX"
  Hmmm ... gss_acquire_cred() happily accepted name "XXXXXXXXXX"
  cred owner resolves to = "SAP_ALL\XXXXXXXXXX"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "CBA/ZYX"
  Hmmm ... gss_acquire_cred() happily accepted name "CBA/ZYX"
  cred owner resolves to = "CBA\ZYX"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "CBA\ZYX"
  Hmmm ... gss_acquire_cred() happily accepted name "CBA\ZYX"
  cred owner resolves to = "CBA\ZYX"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "CBA@ZYX"
  Hmmm ... gss_acquire_cred() happily accepted name "CBA@ZYX"
  cred owner resolves to = "SAP_ALL\CBA@ZYX"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "XY Z"
  Hmmm ... gss_acquire_cred() happily accepted name "XY Z"
  cred owner resolves to = "SAP_ALL\XY Z"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "AB C"
  Hmmm ... gss_acquire_cred() happily accepted name "AB C"
  cred owner resolves to = "SAP_ALL\AB C"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "A B C D"
  Hmmm ... gss_acquire_cred() happily accepted name "A B C D"
  cred owner resolves to = "SAP_ALL\A B C D"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "A@B@C@X"
  Hmmm ... gss_acquire_cred() happily accepted name "A@B@C@X"
  cred owner resolves to = "SAP_ALL\A@B@C@X"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "A@@@@@Z"
  Hmmm ... gss_acquire_cred() happily accepted name "A@@@@@Z"
  cred owner resolves to = "SAP_ALL\A@@@@@Z"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

```

```

Hmmm ... gss_acquire_cred() happily accepted name
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
cred owner resolves to =
"SAP_ALL\XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "a,b,c,d"
Hmmm ... gss_acquire_cred() happily accepted name "a,b,c,d"
cred owner resolves to = "SAP_ALL\A,B,C,D"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "CN=A1,OU=B ,O = C,
C =ES"
Hmmm ... gss_acquire_cred() happily accepted name "CN=A1,OU=B ,O = C, C
=ES"
cred owner resolves to = "SAP_ALL\CN=A1,OU=B ,O = C, C =ES"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "CN=A2, OU=C, O=D, C=ES"
Hmmm ... gss_acquire_cred() happily accepted name "CN=A2, OU=C, O=D, C=ES"
cred owner resolves to = "SAP_ALL\CN=A2, OU=C, O=D, C=ES"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "C=ES, O=C, OU =
B,CN=A3"
Hmmm ... gss_acquire_cred() happily accepted name "C=ES, O=C, OU = B,CN=A3"
cred owner resolves to = "SAP_ALL\C=ES, O=C, OU = B,CN=A3"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "OU=H, C=ES, CN=A4, O=J"
Hmmm ... gss_acquire_cred() happily accepted name "OU=H, C=ES, CN=A4, O=J"
cred owner resolves to = "SAP_ALL\OU=H, C=ES, CN=A4, O=J"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "CN=A5; OU=Y; O=X; C=ES"
Hmmm ... gss_acquire_cred() happily accepted name "CN=A5; OU=Y; O=X; C=ES"
cred owner resolves to = "SAP_ALL\CN=A5; OU=Y; O=X; C=ES"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "CN = A6 , OU=T;O =U, C=
ES"
Hmmm ... gss_acquire_cred() happily accepted name "CN = A6 , OU=T;O =U, C= ES"
cred owner resolves to = "SAP_ALL\CN = A6 , OU=T;O =U, C= ES"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "CN=A7,, OU=Z,O=Y,, ,C=ES"
Hmmm ... gss_acquire_cred() happily accepted name "CN=A7,, OU=Z,O=Y,, ,C=ES"
cred owner resolves to = "SAP_ALL\CN=A7,, OU=Z,O=Y,, ,C=ES"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "CN=A8;; OU=T;O=S;;
;C=ES"
Hmmm ... gss_acquire_cred() happily accepted name "CN=A8;; OU=T;O=S;; ;C=ES"
cred owner resolves to = "SAP_ALL\CN=A8;; OU=T;O=S;; ;C=ES"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "A:B%C/D\E+F-G@H"
Hmmm ... gss_acquire_cred() happily accepted name "A:B%C/D\E+F-G@H"
cred owner resolves to = "A:B%C\D\E+F-G@H"
RESULT OK
TEST: Feeding gss_acquire_cred(ACCEPT) with owner name "A@B:C%D/E\F+G-H"
Hmmm ... gss_acquire_cred() happily accepted name "A@B:C%D/E\F+G-H"
cred owner resolves to = "A@B:C%D\E\F+G-H"
RESULT OK
=====
Testing error handling of context establishment funcions ...
-----
TEST: Testing context establishment with target name "XXXXXXX"
RESULT OK
TEST: Testing context establishment with target name "CBA/ZYX"
RESULT OK
TEST: Testing context establishment with target name "CBA\ZYX"
RESULT OK

```

```

TEST: Testing context establishment with target name "CBA@ZYX"
RESULT OK
TEST: Testing context establishment with target name "X\Z"
RESULT OK
TEST: Testing context establishment with target name "AB C"
RESULT OK
TEST: Testing context establishment with target name "A B C D"
RESULT OK
TEST: Testing context establishment with target name "A@B@C@X"
RESULT OK
TEST: Testing context establishment with target name "A@@@@@Z"
RESULT OK
TEST: Testing context establishment with target name
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
RESULT OK
TEST: Testing context establishment with target name "a,b,c,d"
RESULT OK
TEST: Testing context establishment with target name "CN=A1,OU=B ,O = C, C
=ES"
RESULT OK
TEST: Testing context establishment with target name "CN=A2, OU=C, O=D, C=ES"
RESULT OK
TEST: Testing context establishment with target name "C=ES, O=C, OU = B,CN=A3"
RESULT OK
TEST: Testing context establishment with target name "OU=H, C=ES, CN=A4, O=J"
RESULT OK
TEST: Testing context establishment with target name "CN=A5; OU=Y; O=X; C=ES"
RESULT OK
TEST: Testing context establishment with target name "CN = A6 , OU=T;O =U, C=
ES"
RESULT OK
TEST: Testing context establishment with target name "CN=A7,, OU=Z,O=Y,, ,C=ES"
RESULT OK
TEST: Testing context establishment with target name "CN=A8;; OU=T;O=S;; ;C=ES"
RESULT OK
TEST: Testing context establishment with target name "A:B%C/D\E+F-G@H"
RESULT OK
TEST: Testing context establishment with target name "A@B:C%D/E\F+G-H"
RESULT OK
TEST: Testing context establishment with initiator cred for "XXXXXXXXX"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "CBA/ZYX"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "CBA\ZYX"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "CBA@ZYX"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"

```

```

gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "XY Z"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "AB C"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "A B C D"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "A@B@C@X"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "A@@@@@Z"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "a,b,c,d"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "CN=A1,OU=B ,O =
C, C =ES"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
"Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
"The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "CN=A2, OU=C, O=D,
C=ES"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)

```

```
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
  "Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
  "The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "C=ES, O=C, OU =
B,CN=A3"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
  "Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
  "The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "OU=H, C=ES, CN=A4,
O=J"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
  "Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
  "The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "CN=A5; OU=Y; O=X;
C=ES"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
  "Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
  "The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "CN = A6 , OU=T;O
=U, C= ES"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
  "Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
  "The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "CN=A7,, OU=Z,O=Y,,
,C=ES"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
  "Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
  "The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "CN=A8;; OU=T;O=S;;
;C=ES"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
  "Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
  "The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "A:B%C/D\E+F-G@H"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
  "Miscellaneous Failure"
gss_display_status(0x1300000d,GSS_S_MECH_CODE) =
  "The logon attempt failed"
RESULT OK
TEST: Testing context establishment with initiator cred for "A@B:C%D\E\F+G-H"
Status: gss_accept_sec_context #2() == (GSS_S_FAILURE)
gss_display_status(0x000d0000,GSS_S_GSS_CODE) =
  "Miscellaneous Failure"
```

gss_display_status(0x130000d,GSS_S_MECH_CODE) =
"The logon attempt failed"

RESULT OK
TEST: Testing context establishment with acceptor cred for "XXXXXXXX"
RESULT OK
TEST: Testing context establishment with acceptor cred for "CBA/ZYX"
RESULT OK
TEST: Testing context establishment with acceptor cred for "CBA\ZYX"
RESULT OK
TEST: Testing context establishment with acceptor cred for "CBA@ZYX"
RESULT OK
TEST: Testing context establishment with acceptor cred for "XY Z"
RESULT OK
TEST: Testing context establishment with acceptor cred for "AB C"
RESULT OK
TEST: Testing context establishment with acceptor cred for "A B CD"
RESULT OK
TEST: Testing context establishment with acceptor cred for "A@B@C@X"
RESULT OK
TEST: Testing context establishment with acceptor cred for "A@@@@@Z"
RESULT OK
TEST: Testing context establishment with acceptor cred for
"XX"
RESULT OK
TEST: Testing context establishment with acceptor cred for "a,b,c,d"
RESULT OK
TEST: Testing context establishment with acceptor cred for "CN=A1,OU=B ,O = C,
C =ES"
RESULT OK
TEST: Testing context establishment with acceptor cred for "CN=A2, OU=C, O=D,
C=ES"
RESULT OK
TEST: Testing context establishment with acceptor cred for "C=ES, O=C, OU =
B,CN=A3"
RESULT OK
TEST: Testing context establishment with acceptor cred for "OU=H, C=ES, CN=A4,
O=J"
RESULT OK
TEST: Testing context establishment with acceptor cred for "CN=A5; OU=Y; O=X;
C=ES"
RESULT OK
TEST: Testing context establishment with acceptor cred for "CN = A6 , OU=T;O =U,
C= ES"
RESULT OK
TEST: Testing context establishment with acceptor cred for "CN=A7,, OU=Z,O=Y,,
,C=ES"
RESULT OK
TEST: Testing context establishment with acceptor cred for "CN=A8;; OU=T;O=S;;
;C=ES"
RESULT OK
TEST: Testing context establishment with acceptor cred for "A:B%C/D\E+F-G@H"
RESULT OK
TEST: Testing context establishment with acceptor cred for "A@B:C%D/E\F+G-H"
RESULT OK
=====

Finally ...

*
* GSStest Result Summary :
*

GSSTEST Release : Version 1.05 27-Nov-1998
 built on : Nov 27 1998 at 16:43:43

Command line:
 argv[0] = "gsstest"
 argv[1] = "-l"
 argv[2] = "gssntlm.dll"
 argv[3] = "-o"
 argv[4] = "gssntlm-nt.log"

GSS-API Library : D:\sap\gsstest\debug\gssntlm.dll

 Name = "Builtin SNC-Adapter for SAPntlm SSO (NT4/Win95)"

supported SAP R/3 Releases : 3.1I, 4.0B, 4.5x, 4.6x
 Mech Prefix : "sapntlm"
 SAPGSS_ID : 4
 Support for context import/export : Yes
 Support for mutual authentication : No
 Support for data replay protection : No
 Support for confidentiality protection : No
 Support for integrity protection : No

Private nametype OIDs:
 "p:" = {1 3 6 1 4 1 694 2 1 2 1} NT= GSS_SAP_NT_DOMAIN_USER

=====
 Current Date&Time : Mon, 30-Nov-1998 19:57:42 GMT -02:00
 Operating System : Microsoft Windows NT
 -Release : 4.0 (Build 1381)
 Hardware/Machine : x86 cpu_level=6, cpu_rev=0x0102
 Perf-Index (p-90) : opt= 0.80 dbg= 1.60
 Timer Resolution : 0.009 millisec using "QueryPerformanceCounter()"
 Hostname : P18509
 Current user : D019080
 =====

((1 a))
 Mechanism = {1 3 6 1 4 1 694 2 1 2} MECH= SAPntlm SSO (NT4/Win95)
 = { 10, "\053\006\001\004\001\205\066\002\001\002" }

((1 b))
 Nametype = {1 3 6 1 4 1 694 2 1 2 1} NT= GSS_SAP_NT_DOMAIN_USER
 = { 11, "\053\006\001\004\001\205\066\002\001\002\001" }

((1 c))
 default Initiator name: "SAP_ALL\D019080"
 given Acceptor name: "SAP_ALL\D019080"
 canonical Acceptor name: "SAP_ALL\D019080"

 (default) Acceptor name: "SAP_ALL\D019080"

 ((2 a))
 Performance of names management calls

	min	avg	max	:
gss_import_name()	(1144 calls)	0.01	0.02	0.12 ms
gss_display_name()	(2019 calls)	0.01	0.01	0.05 ms
gss_export_name()	(302 calls)	0.01	0.01	0.02 ms
gss_canonicalize_name()	(1083 calls)	0.01	0.21	28.50 ms

gss_compare_name()	(1039 calls)	0.01	0.01	0.05	ms
gss_release_name()	(4330 calls)	0.01	0.01	0.86	ms

((2 b))

Observed sizes of names:

printable names [15 .. 15] bytes
 exported binary canonical names [35 .. 35] bytes

((3 a))

Performance of credential management calls		min	avg	max	:
gss_acquire_cred() Ini	(185 calls)	0.36	4.38	218.02	ms
gss_acquire_cred() Acc	(169 calls)	0.34	1.88	28.48	ms
gss_inquire_cred() Ini	(38 calls)	0.02	0.19	6.09	ms
gss_inquire_cred() Acc	(24 calls)	0.02	0.03	0.04	ms
gss_release_cred()	(354 calls)	0.06	0.17	1.20	ms

((3 b))

Observed Credentials lifetime(s):

Elapsed real time : 00h 01m 17s
 Initiator credentials lifetime constant at : Indefinite
 Acceptor credentials lifetime constant at : Indefinite

Security context establishment (122 contexts)

((4 a))

Mechanism uses 3-way authentication

((4 b))

Security Context Attribute results:

requested = (MUTUAL,REPLAY,CONF,INTEG,TRANS)
 provided = (TRANS)
 denied = (MUTUAL,REPLAY,CONF,INTEG)

Performance of context establishment calls		min	avg	max	:
gss_init_sec_context() #1	(143 calls)	0.25	0.34	1.79	ms
gss_init_sec_context() #2	(143 calls)	1.56	2.16	39.46	ms
gss_accept_sec_context() #1	(143 calls)	0.24	0.32	2.38	ms
gss_accept_sec_context() #2	(122 calls)	19.93	26.39	95.37	ms

((4 c))

Total context establishment overhead

		min	avg	max	:
gss_init_sec_context()	(122 calls)	1.81	2.49	41.25	ms
gss_accept_sec_context()	(122 calls)	20.17	26.71	97.75	ms
gss_delete_sec_context()	(286 calls)	0.02	0.11	1.70	ms
gss_inquire_context()	(960 calls)	0.01	0.02	0.14	ms

((4 d))

Observed initial lifetimes for established security contexts:

All security contexts were established with lifetime "Indefinite".

((4 e))

Observed token sizes for gss_init_sec_context(): [61 .. 171] bytes
 Observed token sizes for gss_accept_sec_context(): [56 .. 56] bytes

Security context transfer: 288 context transfers, 32 cross-process

((5 a))

Performance of security context transfer		min	avg	max :
gss_export_sec_context()	Ini (192 calls)	0.03	0.08	0.58 ms
gss_export_sec_context()	Acc (160 calls)	0.02	0.10	0.78 ms
gss_import_sec_context()	Ini (192 calls)	0.02	0.02	0.04 ms
gss_import_sec_context()	Acc (160 calls)	0.02	0.02	0.08 ms

((5 b))

Interprocess token sizes for Initiator: [124 .. 124] bytes (192 calls)
 Interprocess token sizes for Acceptor: [124 .. 124] bytes (160 calls)

 Message Protection Services:

((6 a))

NOT available!

((6 b))

Performance of per-message calls		min	avg	max :
gss_context_time()	(224 calls)	0.01	0.01	0.03 ms

((7 a))

Performance of remaining GSS-API calls		min	avg	max :
gss_display_status	(128 calls)	0.02	0.04	0.41 ms
gss_release_buffer	(3167 calls)	0.01	0.01	0.15 ms
gss_release_oid_set	(9 calls)	0.01	0.01	0.01 ms
gss_indicate_mechs	(2 calls)	0.02	147.65	295.28 ms

 Displaying of status information via gss_display_status():

((7 b))

gss_display_status(major) output message size [12 .. 37] avg= 25 chars

 Current limits for SNC-interoperability
 with R/3 Releases 3.1x, 4.0x, 4.5x:

max. length of printable names	=	70	octets
max. length of exported (binary) names	=	126	octets
max. size of MIC token by gss_getmic()	=	128	octets
max. message size increase by gss_wrap()	=	128	octets
max. size of context establishment token	=	25000	octets
max. size of exported context token	=	8000	octets
max. context establishment time	=	1000.0	millisec
max. import/export_sec_context time	=	3.0	millisec
max. CPU time for gss_context_time()	=	1.0	millisec

=====

Passing all API result tests.
 Passing all SAP constraints.

--- Passed --- SAP GSS-API v2 Test (builtin SNC-Adapter)

Mechanism	= {1 3 6 1 4 1 694 2 1 2}	MECH= SAPntlm SSO (NT4/Win95)
Nametype	= {1 3 6 1 4 1 694 2 1 2 1}	NT= GSS_SAP_NT_DOMAIN_USER

Max. data protection level = 1 (Authentication only)

Hardware Platform = Microsoft Windows NT 4.0 (Build 1381)

=====
Done.

3.2 More output for security products with message protection

Since GSSNTLM.DLL supports only unidirectional authentication and no message protection services, only a fraction of the GSS-API tests were performed. Here are snippets from the output of a security product that actually implements mutual authentication and message protection services:

```
*****
***
*** "gssctest" -- GSS-API v2 Shared Library API Test Program ***
***
*** Version 1.05      27-Nov-1998 ***
***
*** This implementation is Copyright (c), 1998 SAP AG Walldorf ***
***
*****
*** This tool may be freely used to test functionality and ***
*** robustness of GSS-API v2 mechanism implemenations ***
*****
*** SAP AG DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, ***
*** INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND ***
*** FITNESS. IN NO EVENT SHALL SAP AG BE LIABLE FOR ANY SPECIAL, ***
*** INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER ***
*** RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ***
*** ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ***
*** ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE ***
*** OF THIS SOFTWARE. ***
*****
```

Timer resolution of QueryPerformanceCounter() is (at least) 0.009 millisc
 1 second passed in 1001.235 millisc.

```
=====
Testing generic gssapi functions ...
=====
Testing credentials management functions ...
=====
Testing names management functions ...
=====
Context establishment functions ...
=====
Message Protection functions ...
-----
TEST: Testing Zero-length messages
RESULT OK
TEST: gss_wrap_size_limit()/gss_wrap()/gss_getmic() token size (increase)
gss_wrap_size_limit() growth predictions:
wrap(conf)= [ 64 .. 64 ] wrap(integ)= [ 64 .. 64 ] bytes
Measuring true gss_wrap/gss_getmic size increase -- PATIENCE
wrap(conf)=[ 48 .. 57 ] wrap(integ)=[ 48 .. 57 ] getmic()=[ 37 .. 37 ]
RESULT OK
TEST: Testing immediate context transfer
RESULT OK
=====
Testing gss_display_status() major_status translations ...
=====
Testing error handling of name management functions ...
=====
Testing error handling of credential management functions ...
=====
Testing error handling of context establishment functions ...
```

Finally ...

```
*****
*
*   GSStest Result Summary :
*
*****

GSSTEST Release   : Version 1.05      27-Nov-1998
                   built on    : Nov 27 1998 at 16:43:43
```

```
=====
Current Date&Time : Mon, 30-Nov-1998  12:44:47  GMT -02:00
Operating System  : Microsoft Windows NT
                   -Release   : 4.0 (Build 1381)
Hardware/Machine  : x86  cpu_level=6, cpu_rev=0x0102
Perf-Index (p-90) : opt= 0.80  dbg= 1.60
Timer Resolution  : 0.009 millisec using "QueryPerformanceCounter()"
Hostname          : P18509
Current user      : D019080
=====
```

```
(( 2 a ))
Performance of names management calls           min      avg      max :
gss_import_name()      ( 634 calls)  0.08    0.15    0.94  ms
gss_display_name()    ( 1297 calls) 0.10    0.13    0.98  ms
gss_export_name()     ( 307 calls)  0.07    0.18   28.79  ms
gss_canonicalize_name() ( 635 calls)  0.05    0.06    0.15  ms
gss_compare_name()    ( 804 calls)  0.07    0.08    1.61  ms
gss_release_name()    ( 2816 calls) 0.04    0.05    0.93  ms
```

```
(( 2 b ))
Observed sizes of names:
printable names          [ 26 .. 26 ] bytes
exported binary canonical names [ 66 .. 66 ] bytes
```

```
(( 3 a ))
Performance of credential management calls      min      avg      max :
gss_acquire_cred() Ini   ( 79 calls)  4.28    6.49   81.46  ms
gss_acquire_cred() Acc   ( 63 calls)  4.49    5.07   18.82  ms
gss_inquire_cred() Ini   ( 17 calls)  0.07    1.15    9.39  ms
gss_inquire_cred() Acc   (  3 calls)  0.20    0.25    0.32  ms
gss_release_cred()      ( 142 calls) 0.01    0.02    0.03  ms
```

```
(( 3 b ))
Observed Credentials lifetime(s):
                                     Elapsed real time : 00h 01m 18s

Initiator credentials      lifetime decrease : 00h 01m 18s
Initiator credentials initial lifetime      : 154d 01h 16m 06s
Initiator credentials final   lifetime      : 154d 01h 14m 48s

Acceptor credentials      lifetime decrease : 00h 01m 18s
Acceptor credentials initial lifetime      : 154d 01h 16m 06s
Acceptor credentials final   lifetime      : 154d 01h 14m 48s
```

Security context establishment (52 contexts)

((4 a))

Mechanism uses 3-way authentication

((4 b))

Security Context Attribute results:

```

requested = (MUTUAL,REPLAY,CONF,INTEG,TRANS)
provided  = (MUTUAL,REPLAY,SEQUENCE,CONF,INTEG,TRANS)
donated   = (SEQUENCE)

```

Performance of context establishment calls		min	avg	max :
gss_init_sec_context()	#1 (58 calls)	10.24	11.89	67.21 ms
gss_init_sec_context()	#2 (52 calls)	27.77	30.43	128.40 ms
gss_accept_sec_context()	#1 (58 calls)	32.03	33.76	50.58 ms
gss_accept_sec_context()	#2 (52 calls)	0.87	0.99	2.22 ms

((4 c))

Total context establishment overhead		min	avg	max :
gss_init_sec_context()	(52 calls)	38.01	42.32	195.61 ms
gss_accept_sec_context()	(52 calls)	32.90	34.76	52.80 ms
gss_delete_sec_context()	(116 calls)	0.04	0.07	0.09 ms
gss_inquire_context()	(738 calls)	0.16	0.20	1.78 ms

((4 d))

Observed initial lifetimes for established security contexts:

All security contexts were established with lifetime "Indefinite".

((4 e))

Observed token sizes for gss_init_sec_context(): [44 .. 536] bytes

Observed token sizes for gss_accept_sec_context(): [670 .. 670] bytes

Security context transfer: 234 context transfers, 34 cross-process

((5 a))

Performance of security context transfer		min	avg	max :
gss_export_sec_context()	Ini (135 calls)	0.19	0.23	0.89 ms
gss_export_sec_context()	Acc (168 calls)	0.21	0.25	0.51 ms
gss_import_sec_context()	Ini (135 calls)	0.12	0.14	1.34 ms
gss_import_sec_context()	Acc (168 calls)	0.13	0.15	0.16 ms

((5 b))

Interprocess token sizes for Initiator: [172 .. 172] bytes (135 calls)

Interprocess token sizes for Acceptor: [266 .. 266] bytes (168 calls)

Message Protection Services:

((6 a))

Confidentiality and Integrity

((6 b))

Performance of per-message calls		min	avg	max :
gss_context_time()	(2049 calls)	0.01	0.01	1.02 ms
gss_wrap_size_limit()	(7654 calls)	0.01	0.01	0.50 ms

((6 c))

GSS-API message protection throughput		avg	max :
gss_getmic()	(1047 calls)	1.2	5.3 KByte/msec min= 0.74 ms
gss_verifymic()	(1048 calls)	1.2	5.4 KByte/msec min= 0.73 ms
gss_wrap(mic)	(883 calls)	1.1	5.0 KByte/msec min= 0.74 ms
gss_unwrap(mic)	(883 calls)	1.2	5.0 KByte/msec min= 0.74 ms
gss_wrap(conf)	(1122 calls)	0.4	0.6 KByte/msec min= 0.75 ms

gss_unwrap(conf) (1122 calls) 0.4 0.7 KByte/msec min= 0.75 ms

((6 d))

Token sizes for gss_getmic() : [37 .. 37] bytes
 Message size increase for gss_wrap(mic) : [48 .. 57] bytes
 Message size increase for gss_wrap(conf): [48 .. 57] bytes

((6 e))

non-default QOP return values for gss_getmic():
 0x00000002
 non-default QOP return values for gss_wrap(mic):
 0x00000002
 non-default QOP return values for gss_wrap(conf):
 0x00000002

((7 a))

Performance of remaining GSS-API calls		min	avg	max :
gss_display_status	(100 calls)	0.02	0.13	10.98 ms
gss_release_buffer	(10036 calls)	0.01	0.01	1.04 ms
gss_release_oid_set	(9 calls)	0.01	0.01	0.02 ms
gss_indicate_mechs	(2 calls)	0.79	42.18	83.57 ms

=====

Done.