# Contents

*What's New in SAS/ACCESS 9.4 Interface to R/3* .............................................. v

**Chapter 1 / Getting Started with SAS/ACCESS Interface to R/3** .......................... 1
  - Getting Started with SAS/ACCESS Interface to R/3 ........................................ 1
  - Searching R/3 Metadata .................................................................................. 3
  - Exporting R/3 Metadata to SAS/Warehouse Administrator ................................ 5
  - Window and Field Help .................................................................................. 7

**Chapter 2 / SAP Data Types for SAS/ACCESS Interface to R/3** ......................... 9
  - SAP Data Types for SAS/ACCESS Interface to R/3 ........................................... 9

**Chapter 3 / Accessing and Using R/3 Data** ...................................................... 11
  - Accessing Data Using Direct Access and ODBC ............................................. 11
  - Accessing Data Using SAS/ACCESS Interface to R/3 ..................................... 12
  - Accessing Data Using Secure Network Communications ............................... 14
  - Authorization Checks for BW and BI Objects .................................................. 15

**Chapter 4 / Using the SAS RFC Server** ............................................................. 17
  - Overview of the SAS RFC Server ................................................................... 17
  - Using the SAS RFC Server on UNIX ............................................................... 17
  - Using the SAS RFC Server on Windows ......................................................... 19
  - Using the SAS RFC Server in Batch Mode ..................................................... 21

**Chapter 5 / Using the RFC Macros and Macro Variables** ............................... 23
  - Using the RFC Macros and Macro Variables .................................................. 23
  - Using the R/3 BAPI Connector: Logon Window ............................................. 24
  - Performing Batch Operations ......................................................................... 25
  - Dictionary ...................................................................................................... 28

**Chapter 6 / Using the LIBNAME Statement for SAP** ....................................... 41
  - Overview of the LIBNAME Engine for SAP ..................................................... 42
  - LIBNAME Statement Syntax for SAP .............................................................. 42
  - SAP LIBNAME Engine Options ...................................................................... 43
  - SAS LIBNAME Statement Options for the SAP Engine ................................. 44
  - SAS Data Set Options for the SAP Engine ..................................................... 45
  - Dictionary ...................................................................................................... 45
  - Examples: ....................................................................................................... 67

**Chapter 7 / CALLRFC Procedure** .................................................................. 69
  - Overview: CALLRFC Procedure .................................................................. 69
  - Syntax: CALLRFC Procedure ....................................................................... 69
  - Usage: CALLRFC Procedure ....................................................................... 79
  - Examples: CALLRFC Procedure .................................................................. 80

**Appendix 1 / Support for SAP Basic Objects** ................................................... 83
  - SAP Basic Objects Supported by SAS/ACCESS Interface to R/3 ................. 83
  - SAP Basic Objects Supported by SAS Data Surveyor for SAP ...................... 83
Objects Supported in an SAP BW on an SAP HANA System .................. 84
What's New in SAS/ACCESS 9.4 Interface to R/3

Overview
SAS/ACCESS Interface to R/3 provides a read-only engine for using SAS to access data in your SAP R/3 or SAP BW system. SAS/ACCESS Interface to R/3 has these changes and enhancements:

- new LIBNAME options
- new CALLRFC options
- SAS Viya 3.4 supports the R/3 engine, but does not have a data connector.
- In SAS Viya 3.4, SCHEMA LIBNAME option and NAMESPACE LIBNAME option are now available for all customers.

New LIBNAME Options
In SAS 9.4M3, two new LIBNAME options are available:

- The CLASSIC option specifies that the engine uses the classic RFC libraries.
- The NETWEAVER option specifies that the engine uses the SAP NetWeaver RFC libraries.

New CALLRFC Procedure Options
In SAS 9.4M3, two new CALLRFC procedure options are available:

- The CLASSIC option specifies that the engine uses the classic RFC libraries.
- The NETWEAVER option specifies that the engine uses the SAP NetWeaver RFC libraries.
Documentation Enhancement

In November 2019, a chapter was added entitled "SAP Data Types for SAS/ACCESS Interface to R/3".

The SAP basic objects supported by SAS/ACCESS Interface to R/3 and SAS Data Surveyor for SAP are now included. The new topic also includes the objects supported in an SAP BW on an SAP HANA system.
Getting Started with SAS/ACCESS Interface to R/3

This section provides a brief tutorial on how to use SAS/ACCESS Interface to R/3. For details, see the additional references in the “Recommended Reading” section at the front of this document or contact your on-site SAS support personnel.

Follow these steps to get started with SAS/ACCESS Interface to R/3:

1. Open a SAS session.
2. Enter `%r3access` in the command line to start SAS/ACCESS Interface to R/3:

   ![SAS Command Line]

   Figure 1.1  SAS Command Line

The SAS/ACCESS to R/3 desktop appears.
Figure 1.2  SAS/ACCESS to R/3 Desktop

Double-click the Logon icon to open the Logon to R/3 window.

Figure 1.3  Logon to R/3 Window

Existing predefined profiles are available from the Profile field drop-down list.

Follow these steps to use a predefined profile.

1 Select a profile from the Profile field drop-down list.
2 Click OK to connect to the SAP system.

If no predefined profiles exist, you must create one.

Follow these steps to create a new profile.

1 Open the Logon to R/3 window as described earlier.
2 If you want to save the connection information to use later, enter a name for the Profile.
3 Enter a three-digit identification number for the Client.

4 Enter a valid User ID.

5 Enter the corresponding Password for the user ID.

6 Click Advanced in the Logon to R/3 window to display the Advanced Parameters window.

Figure 1.4 Advanced Parameters Window

7 For z/OS operating environments, follow these steps.
   a Select the TCP/IP tab and enter the SAS RFC Server name in the Host field. The default is localhost. A fully qualified domain name might not be required, depending on the DNS configuration for the network.
   b Enter a valid port number in the Port field.

8 From the RFC tab, enter the SAP system host name in the Host field.

9 Click OK to close the Advanced Parameters window and return to the Logon to R/3 window.

10 If you want to use this logon information later, click Save to save the profile information.

11 Click OK to connect to the SAP system.

Searching R/3 Metadata

SAS/ACCESS Interface to R/3 provides a robust search capability that you can use to search your metadata for specific information.
For example, to search for all tables that contain the word *scrap* in their metadata.

1. From SAS, start SAS/ACCESS Interface to R/3 to display the SAS/ACCESS to R/3 window.

2. Double-click the Search Metadata icon to display the Data Dictionary Search Facility window:

![Data Dictionary Search Facility Window](image)

3. Click inside the text entry field on the **Search String** tab. Enter text to search for, such as the word "scrap." Click **search**. A list of tables is displayed in the table in the bottom of the **Search String** tab, as shown below.

![Data Dictionary Search Facility Window (Searching for `scrap`)](image)

4. Double-click a table in the list. The General Attributes for a List of Tables window displays to provide more detailed information about the selected table.
5 Click OK to return to the Data Dictionary Search Facility window.

## Exporting R/3 Metadata to SAS/Warehouse Administrator

With SAS/ACCESS Interface to R/3, you can export metadata from SAP tables to a warehouse environment in SAS/Warehouse Administrator.

Follow these steps to export metadata.

1. From SAS, open SAS/ACCESS Interface to R/3 to display the SAS/ACCESS to R/3 desktop.

2. Double-click the Datamodel Explorer icon to display the Datamodel Explorer window.
3 For this example, click the plus sign (+) for the application data models topic in the Datamodel Explorer. Then, click the plus sign (+) beside the Industry Solutions and Industry Solutions Public Sector subtopics. Click ISPS_HR IS-PS Human Resources to select this subtopic.

4 From the main menu of the Datamodel Explorer window, select File ➤ Export Metadata ➤ SAS/Warehouse Administrator to display the SAS/ACCESS to R/3: Export Metadata to SAS Warehouse Administrator window.

5 From the Data Warehouse Environment drop-down list, select a predefined environment from the SAS/ACCESS to R/3 folder.
6 From the Operational Data Group drop-down list, select an operational data group from the selected data warehouse environment. The Operational Data Group field is disabled until you select a data warehouse environment.

7 Click OK to export the metadata to the selected data warehouse environment in SAS/Warehouse Administrator.

Note: SAS Viya 3.4 supports the R/3 engine, but does not have a data connector.

Here are the general steps to manage and manipulate your data in SAS Viya without an associated data connector.

1 Perform any necessary data preparation before loading data into the SAS client. For example, if you want to load the join result of two Hive tables into CAS, create a Hive table that is the result of the join. You can then load the resulting table into CAS.
   Perform the data preparation by interacting directly with your data source or by using the SQL procedure in SAS.

2 Establish a connection between the SAS client and your database by defining a libref in a LIBNAME statement. You can then load your data into the SAS client using a DATA step.

3 Establish a connection between the SAS client and a CAS session by using a CAS LIBNAME statement. You can then load your data into a CAS table on the CAS server.

4 Perform analysis on your data in a CAS session. You can use the DATA step, procedures that work with CAS, and CAS actions on your CAS table. For more information, see SAS Viya Quick Start and SAS Cloud Analytic Services: Fundamentals.

Window and Field Help

For descriptions of windows and buttons in the SAS/ACCESS to R/3 desktop, click Help on any window to access SAS/ACCESS Interface to R/3: Help.
Below is a list of the ABAP data types for SAP that are supported for SAS/ACCESS Interface to R/3. For more detailed information, see your SAP documentation.

**Table 2.1 SAP Data Types**

<table>
<thead>
<tr>
<th>DATA Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCP</td>
<td>Posting period with the format of YYYYMM in the form <strong>.</strong>.</td>
</tr>
<tr>
<td>CHAR</td>
<td>Character string</td>
</tr>
<tr>
<td>CLNT</td>
<td>Client</td>
</tr>
<tr>
<td>CUKY</td>
<td>Currency key</td>
</tr>
<tr>
<td>CURR</td>
<td>Currency</td>
</tr>
<tr>
<td>DATS</td>
<td>Date</td>
</tr>
<tr>
<td>DEC</td>
<td>Counter with decimal point, sign, and comma for thousands</td>
</tr>
<tr>
<td>FLTP</td>
<td>Floating-point number including decimals</td>
</tr>
<tr>
<td>INT1</td>
<td>1-byte integer with a value range of 0 to 255</td>
</tr>
<tr>
<td>INT2</td>
<td>2-byte integer</td>
</tr>
<tr>
<td>DATA Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>INT4</td>
<td>4-byte integer</td>
</tr>
<tr>
<td>LANG</td>
<td>Language key</td>
</tr>
<tr>
<td>LCHR</td>
<td>Character string</td>
</tr>
<tr>
<td>LRAW</td>
<td>Uninterpreted byte string</td>
</tr>
<tr>
<td>NUMC</td>
<td>Character string containing only digits</td>
</tr>
<tr>
<td>PREC</td>
<td>Precision of a QUAN field</td>
</tr>
<tr>
<td>QUAN</td>
<td>Quantity</td>
</tr>
<tr>
<td>RAW</td>
<td>Uninterpreted byte string</td>
</tr>
<tr>
<td>TIMS</td>
<td>Time with the format HHMMSS in the form <strong>.</strong>.__.</td>
</tr>
<tr>
<td>UNIT</td>
<td>Units key</td>
</tr>
<tr>
<td>VARC</td>
<td>Character string of variable length. This is now an obsolete data type.</td>
</tr>
</tbody>
</table>
Accessing Data Using Direct Access and ODBC

Here are the ways that you can access data that does not require SAS/ACCESS Interface to R/3.

Direct Access
This is a form of transparent access that does not require additional programming or intermediate operations. This form of access is possible because the SAP system is an open systems-compliant application. Direct access uses an intermediate SAS/ACCESS product to the specific database.

ODBC (Open Database Connectivity)
This method allows direct data access in a PC environment by using SAS/ACCESS Interface to ODBC.
Accessing Data Using SAS/ACCESS Interface to R/3

Overview

You can use SAS/ACCESS Interface to R/3 to access SAP data two different ways. Both methods generate a SAS DATA step.

- Remote Function Call on page 12. This typical method offers online access to the SAP data.
- Extract Management on page 13. Now largely superseded, this original method offers offline access to SAP data. It can be most useful in these circumstances:
  - A poor or slow communication link exists between SAS and SAP.
  - A backup file is required.
  - There is a lot of data.

Remote Function Call (RFC) Method

The Remote Function Call (RFC) is an SAP implementation of the Remote Procedure Call. RFC allows access to R/3 from external applications, such as SAS or other SAP systems.

An overview of the Remote Function Call method for accessing data from SAP is shown in the following figure. Two modules (ABAP Function and ABAP Report, which are provided with the interface) attach to the R/3 system. Communication between the ABAP Function module and the interface itself uses RFC over a TCP/IP connection. In addition to the components in the previous figure, in a z/OS environment a SAS RFC Server is deployed between the SAP system and the SAS Server to provide access to data in the SAP system.
In general, the RFC method consists of generating a SAS DATA step to access the SAP data. It then uses the DATA step to create one of these items:

- a data set that is a SAS copy of the SAP data
- a view that reads the SAP data each time it is required

**Extract Management Method**

**Overview**

The extract management method creates an ABAP query that copies the data from SAP to an external file. It also creates a SAS DATA step that then reads the external file into SAS. After the SAP data has been exported to SAS, the data can be held either temporarily for discrete analysis or permanently in a SAS data warehouse. The interface generates all of the necessary programs for both ABAP and SAS (based on the SAP data dictionary tables).

The extract management method to access SAP data is discussed in the following sections.

**Installation and Setup**

1. Generate an ABAP report.
Use the descriptor file to create an ABAP report and write it to a text file. The ABAP report is used later to copy the SAP data to an external intermediate data file.

2 Generate a DATA step.
   Generate a SAS DATA step to read the intermediate data file.

3 Install the ABAP report in R/3.
   This step is independent of SAS/ACCESS Interface to R/3 and SAS and can be performed simultaneously with Step 2.

4 Install the ABAP text file from Step 1 into SAP.

Data Transfer and Use

Note: After installation and setup, you can perform the following steps as many times as required.

1 Run the ABAP report in R/3.
   Run the installed ABAP report on SAP. This creates the external intermediate data file.

2 Use the DATA step.
   Use the DATA step to read in or copy the external intermediate data file into SAS.

Note: Depending on your situation, it might be possible to simplify or streamline the outlined process. For example, in UNIX a named pipe or other external file can enable ABAP to write data to the external intermediate data file so that SAS can read it simultaneously.

Accessing Data Using Secure Network Communications

Overview

The SAP system and the RFC protocol support secure network communications (SNC). You can configure SNC to enable single sign-on (SSO). Although you can also use SNC without single sign-on, you must still include user name and password credentials. This section shows two examples of how to specify the SNC logon information with the SAP LIBNAME engine. You must configure the SAP system and user accounts for SNC. For information about how to do this, see the SAP document, Secure Network Communications User’s Guide.
Using the saprfc.ini File

This method requires that you create an entry in the saprfc.ini file. Here is an example:

```
DEST=SI9
TYPE=A
ASHOST=sapr3srv.sup.com
SYSNR=03
SNC_MODE=1
SNC_PARTNERNAME=p:SAPServiceSI9@sup.com
SNC_LIB=C:\Public\win32sso\gssntlm.dll
```

Then, use the LIBNAME statement to use the connection:

```
libname a r3 dest=SI9;
```

Using the RFC_STRING Option

You can also use the RFC_STRING option to apply additional settings:

```
libname a r3 ashost=sapr3srv.sup.com sysnr=03
rfc_string="SNC_MODE=1 SNC_PARTNERNAME=p:SAPServiceSI9
SNC_LIB=C:\Public\win32sso\gssntlm.dll";
```

The SNC_LIB string must not contain blanks.

This document shows two examples of how to use SNC for single sign-on in a LIBNAME statement. For more information about SNC and SNC configuration, see SAP documentation and other third-party documentation.

Authorization Checks for BW and BI Objects

The SAP system release level determines how authorization checks are performed for BW and BI objects. In releases before SAP BI 7.0, SAP uses the reporting authorization that uses the SAP standard authorization concept. For BI 7.0 and later, SAP uses an authorization concept for analysis authorization. SAS/ACCESS Interface to R/3 delivers one Business Add-In (BADI) that implements the reporting authorization, and one BADI that implements analysis authorization.

During installation, SAS is configured to use one of three implementations for table access authorization checks. The default implementation when no BADI is activated uses the SAP authorization object S_TABU_DIS. If you want to use one of the two BADI implementations, you must activate it during installation. Here is a description of the implementations.
<table>
<thead>
<tr>
<th>Authorization Check Implementation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Default, no BADI)</td>
<td>$S_TABU_DIS$ authorization object</td>
</tr>
<tr>
<td>Classic BADI /SAS/AUTHBW01</td>
<td>Compatible with BW and BI systems. User authorization checks are performed at the InfoCube, InfoObject, and ODS levels using the reporting authorization (SAP standard authorization concept).</td>
</tr>
<tr>
<td>Enhanced BADI /SAS/IM_AUTHBI01</td>
<td>Compatible with BI 7.0 and later systems. User authorization checks are performed using the analysis authorization. This not only provides an authorization check for the infoProvider (infoCube, infoObject, and DSO) but also column-level restrictions on master data attributes and key figures, and row-level restrictions on attributes.</td>
</tr>
</tbody>
</table>
Overview of the SAS RFC Server

The SAS RFC Server is part of SAS/ACCESS Interface to R/3 in a z/OS operating environment. It is a component of the SAP LIBNAME engine that provides access to the data in SAP systems, supports calling ABAP RFC functions, and uses the TCP/IP protocol for communication with the SAP LIBNAME engine. For more information about the SAP LIBNAME engine, see "Overview of the LIBNAME Engine for SAP" on page 42.

Here is important information to keep in mind about the SAS RFC Server.

- It runs independently of SAS as a background process on the SAS host, the R/3 host, and another machine entirely.
- You can configure it to listen on any TCP/IP port (default 6999).
- You can have more than one concurrent instance of it on a given machine, but each instance must have a unique port number. Each instance can communicate with many SAP systems.
- It can process requests from multiple SAP application servers.
- It can support Unicode and non-Unicode SAP systems.

Using the SAS RFC Server on UNIX

In a UNIX environment, the SAS RFC Server for SAP is installed in the $SASROOT/saspgm/dbi/bin directory. You must add this directory to the PATH environment variable and add the location of the shared libraries to the appropriate environment variable. For details about how to set up environment variables, see Post-Installation Instructions for SAS/ACCESS Interface to R/3.
To start the SAS RFC Server in a UNIX environment, use this command:

```
```

Optional parameters are enclosed in brackets [].

To stop the SAS RFC Server in a UNIX environment, use this command:

```
sasrfc_server [-k] [-p port]
```

Optional parameters are enclosed in brackets [].

Here are the options for the SAS RFC Server command for UNIX.

- **-d n**
  specifies whether to run the server as a daemon process or as an application and specifies the diagnostic level. The server is run as a daemon process when you specify `-d n` (where `n=0`) or when you do not specify the `-d` option. After successful initialization, all messages are routed to `syslog`. When you specify the `-d n` option (where `n= any number between 1 and 6`), the program runs as an application and messages are routed to standard error. Higher diagnostic levels route more detailed messages to standard error.

- **-k**
  stops the service.

- **-n name**
  specifies the name of the service. This option distinguishes among multiple daemon processes. For example, a test service might be started as `-n test`.

- **-p port**
  identifies the port number on which the service listens. The default is 6999.

- **-w workdir**
  specifies the working directory. For example, if your working directory is named `tmp`, use `-w /tmp` to specify this option.

- **-N namespace**
  specifies the namespace for ABAP functions and programs that SAS uses. The default is `/SAS/`. If the ABAP programs are installed in the customer namespace instead of in the `/SAS/` namespace, this parameter identifies where ABAP programs are installed.

- **-V variant**
  specifies the report variant that is assigned to this server. You must assign a report variant for each SAS RFC Server that accesses an SAP application server in batch mode. The variant parameter G_DEST specifies a unique destination to call back to the SAS RFC Server.

- **-help**
  specifies that you want to display command line Help.

- **-c dialog_compat_port**
  identifies the option that, when specified, runs the SAS RFC Server in SAS 8 compatibility mode on the port indicated by the `dialog_compat_port` parameter. The port number must be different from the one used for the SAS 9 requests. If you omit the `-c` option, no dialog compatibility is available.

- **-b"-p port_number [optional] -V variant_name [required] -R report_name" [optional]**
  enables batch compatibility so that you can process SAS 8 ACCR3 requests while you are using a SAS 9 RFC Server. You must use double quotation marks
in the batch compatibility option to separate batch compatibility parameters from SAS 9 parameters that use the same syntax. If you omit the port parameter, the port number defaults to 6999. The report variant parameter must be a valid variant of the report. The values in port_number, variant_name, and report_name must be different from the values that SAS 9 and dialog compatibility uses.

-B number_of_bytes
sets the minimum buffer size for data transfers in batch and dialog modes. The number_of_bytes should be greater than 10,000 and no more than 8 digits, or it is ignored. If you omit the -B option, the default is 10,000 bytes. In batch mode, if you specify a value for the G_BUFMAX parameter that is less than the value of -B, the -B value is used.

For example, to start the SAS RFC Server as a daemon process with the name of test on port number 6991, use this command:

```
sasrfc_server -n test -p 6991
```

This sample command shows how to display command line Help:

```
sasrfc_server
```

To start the SAS RFC Server with debug level 4 on port number 6991, use this command:

```
sasrfc_server -d 4 -p 6991
```

To stop a SAS RFC Server that is running on port number 6991, use this command:

```
sasrfc_server -k -p 6991
```

Using the SAS RFC Server on Windows

In a Windows environment, the SAS RFC Server for SAP is installed in the %SASROOT%\access\sasexe directory.

To start or stop the SAS RFC Server using a command line in the Windows operating environment, use this command:

```
sasrfc_server [-i n] [-r] [-s] [-t n] [-d n] [-n name[,desc]] [-p port]
[-N namespace] [-V variant] [-help] [-c dialog_compar_port]
[-b "-p port-number -V variant_name -R report_name"]
[-B number_of_bytes]
```

Here are the options for the SAS RFC Server command for Windows.

-i n
installs the program as a Windows service and specifies the start-up option for a new service. Here are the start-up options.

- n=2
starts the service automatically

- n=3
starts the service on demand

- n=4
disables the service
-r
removes the program from the Service Manager.

-s
starts the service.

-k
stops the service

-t n
sets the start-up option for an existing service. Here are the start-up options.
   n=2
   starts the service automatically
   n=3
   starts the service on demand
   n=4
   disables the service

-d n
specifies the diagnostic level. When you specify the -d n option where n= any number between 1 and 6, the program runs as an application and messages are routed to standard error. Higher diagnostic levels route more detailed messages to standard error.

-n name, desc
specifies the name and description of the service.

-p port
specifies the port number on which the server listens. The default is 6999.

-N namespace
specifies the namespace for ABAP functions and programs that SAS uses. The default is /SAS/. If the ABAP programs are installed in the customer namespace instead of in the /SAS/ namespace, this parameter identifies where the ABAP programs are installed.

-V variant
specifies the R/3 report variant that is assigned to this server. You must assign a report variant for each SAS RFC Server that accesses an R/3 application server in batch mode. The variant parameter G_DEST specifies a unique destination to call back to the SAS RFC Server.

-help
specifies that you want to display command line Help.

-c dialog_compat_port
identifies the option that, when specified, runs the SAS RFC Server in SAS 8 compatibility mode on the port indicated by the dialog_compat_port parameter. The port number for a SAS 9.1 request must be different from the one that was used for the SAS 9 requests. If you omit the -c option, no dialog compatibility is available.

-b "-p port_number [optional] -V variant_name [required] -R report_name" [optional] enables batch compatibility so that you can process SAS 8 ACCR3 requests while you are using a SAS 9 RFC Server. You must use double quotation marks in the batch compatibility option to separate batch compatibility parameters from SAS 9 parameters that use the same syntax. If you omit the port parameter, the port number defaults to 6999. The report variant parameter must be a valid variant of the report. If you omit the report parameter, the report name defaults to /SAS/ZSASV8COMPAT.
-B number_of_bytes
sets the minimum buffer size for data transfers in batch and dialog modes. The number_of_bytes should be greater than 10,000 and no more than 8 digits, or it is ignored. If you omit the -B option, the default is 10,000 bytes. In batch mode, if you specify a value for the G_BUFMAX parameter that is less than the value of -B, the -B value is used.

For example, if you want to start the SAS RFC Server with a debug level of 1 on port number 6991, use this command:

```
sasrfc_server -d 1 -p 6991
```

To install the SAS RFC Server as a service with automatic start up on port 6991 and use the name test and a description of 'Test RFC Server', use this command:

```
sasrfc_server -i 2 -n "test,Test RFC Server" -p 6991
```

To uninstall the server that was started in the previous example, use this command:

```
sasrfc_server -r -n "test"
```

Using the SAS RFC Server in Batch Mode

The SAS RFC Server can perform batch mode processing of RFC calls to the SAP system. Users who submit interactive RFC calls to the SAP system might encounter time-out limitations. This occurs when interactive SAP jobs exceed the default processing time-out limit of 10 minutes.

By allowing batch mode processing of RFC calls to the SAP system, time-out limitations for interactive SAP dialog processes are resolved. Batch mode processing is often used to reduce the use of dialog processes. It is also used to reduce the impact of extractions—a SAS program reading the tables and views of the SAP system by means of the SAP LIBNAME engine—on the SAP system. Batch processes usually run at a lower priority. Batch mode processes of RFC calls can also help control the number of simultaneous extractions.

The ABAP program that is normally generated (by another ABAP program) executes a requested read and join operation on one or more database tables. When operating in batch mode, the SAS RFC Server creates and releases an SAP job to execute this ABAP program. The SAS RFC Server then waits for the job that is running the ABAP program to start and to recall the SAS RFC Server. Finally, the resulting data stream (that the SAS RFC Server receives from the SAP system) is passed on to the SAP LIBNAME engine.

LIBNAME or data set option: BATCH | BATCH_MODE | BATCHMODE= 0|1|Y|N
Indicates whether the SAS RFC Server should use SAP batch jobs for the data extracts.

Y
RFC uses batch jobs to extract R/3 data

N
RFC uses dialog processes to extract R/3 data

The default value for this option is N.

To submit batch requests to the SAS RFC Server, you must set up an RFC destination and a variant for the /SAS/Z_SAS_READ ABAP program for each SAS
RFC Server instance. Before the SAS RFC Server can execute requests as a batch job, follow these steps.

1 Create a SAS RFC Server destination. For each SAS RFC Server that accesses an SAP system, a destination must be set up on that SAP system. The program ID and the gateway information defined in the destination are used to register the SAS RFC Server on an SAP gateway. If you use two SAS RFC Servers to read data from one SAP application server, you must set up two RFC destinations on that SAP application server. If you omit gateway information from the SM59 destination, the batch job can run only on the R/3 server that is defined in the SAS LIBNAME statement for the request. If you enter explicit gateway information in the SM59 destination information, the batch job runs on any eligible R/3 server as determined by the SAP site batch configuration. Omitting gateway information is typically used in small SAP configurations such as test, QA, and single-server systems. In production environments, it is preferable that jobs be run on a server that the SAP batch system allocates.

2 Create a variant of ABAP program /SAS/Z_SAS_READ. After defining the destination for the SAS RFC Server, you need to make the RFC destination known to the ABAP program by creating a variant for the ABAP program. The concept of variants provides a way of parameterizing ABAP programs. For each defined destination, you must create a variant that references the destination. Variants are client-dependent.

3 Start the SAS RFC Server with the variant. For more information about using the SAS RFC Server and particularly the sasrfc_server option, see “Using the SAS RFC Server on UNIX” on page 17 and “Using the SAS RFC Server on Windows” on page 19.

For details about setup of SAS RFC Server destinations and variants, see Post-Installation Instructions for SAS/ACCESS Interface to R/3.
Using the RFC Macros and Macro Variables

SAS/ACCESS Interface to R/3 includes several SAS macros and macro variables that let you communicate with the SAS RFC Server. Here are the available SAS macros and macro variables.

<table>
<thead>
<tr>
<th>Macro or Macro Variable</th>
<th>Purpose</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>%CALLRFC</td>
<td>Calls a specified function.</td>
<td>“%CALLRFC Autocall Macro” on page 28</td>
</tr>
<tr>
<td>%R3CONNBB</td>
<td>Uses a profile to log on to the SAP system for batch processing.</td>
<td>“%R3CONNBB Autocall Macro” on page 32</td>
</tr>
<tr>
<td>%R3CONNNC</td>
<td>Logs on to the SAP system when performing batch processing.</td>
<td>“%R3CONNNC Autocall Macro” on page 34</td>
</tr>
</tbody>
</table>
### Using the R/3 BAPI Connector: Logon Window

This window is displayed automatically if you do not use the RFC_LOGON_INFO macro string when you submit your %CALLRFC macro statement. Use it to define the R/3 BAPI connection parameters that are required to access the SAP system.

**Figure 5.1  R/3 BAPI Connector: Logon Window**

- **Client**: lets you identify the SAP logon client.
- **User**:
- **Password**:
- **Language**:
- **Destination**:
- **Host**:
- **System number**:
- **Gateway host**:
- **Gateway service**:

For more information about the %CALLRFC macro, see “%CALLRFC Autocall Macro” on page 28. For more information about using the RFC_LOGON_INFO macro string, see “RFC_LOGON_INFO Automatic Macro Variable” on page 37.

This window contains these fields:

- **Client** [required]
  - lets you identify the SAP logon client.
User [required]
lets you identify the SAP user logon ID.

Password [required]
lets you enter the SAP user password.

Language [required]
lets you specify the SAP logon language. Specify the 1-byte SAP language (E for English, D for German, and so on) or the 2-byte ISO language (EN for English, DE for German, and so on).

Destination
lets you identify the logical destination of the saprfc.ini file, if applicable. This field is required if you use the saprfc.ini file.

Host
lets you identify the application server host. This field is required if you use a specific application server.

System number
lets you identify the SAP system number if you use a specific application server and are not using Load Balancing. This field is required if you use a specific application server.

Gateway host
lets you identify the host name of the SAP gateway.

Gateway service
lets you identify the service of the SAP gateway.

The R/3 BAPI Connector Logon window contains these buttons:

OK
submits the R/3 BAPI connection parameters that you entered and tries to log on to the SAP system.

Cancel
closes the R/3 BAPI Connector: Logon window without submitting the connection parameters and trying to log on to the SAP system.

Help
displays the SAS Help for the window.

Performing Batch Operations

Overview

You can use batch operations to automate processes. For example, you can use them for overnight operations or to simplify connections for end users.

Use this command open the connection for a batch operation:

```bash
%r3connb(profile=profile-name,
libref=SAS-libref, function=OPEN);
```
profile-name
specifies the name of the profile to use to save the information in the Logon to R/3 window.

SAS-libref
specifies the SAS library where the profile was saved. If you do not specify this parameter, the profile is sought in the WORK, SASUSER, and R3LIB SAS libraries, in that order.

function=OPEN
opens the connection.

Use one of these commands to close the connection:

$r3connb(profile=profile-name,
libref=SAS-libref,
function=CLOSE);

$r3connb(id=connection-id,
function=CLOSE);

profile-name
specifies the name of the profile to use to save the information in the Logon to R/3 window.

SAS-libref
specifies the SAS library where the profile was saved. If you do not specify this parameter, the profile is sought in the WORK, SASUSER, and R3LIB SAS libraries, in that order.

function=CLOSE
disconnects from the SAP system.

connection_id
specifies the connection ID.

Source Parameter

Previous versions of SAS/ACCESS Interface to R/3 included a source parameter in the batch connection. For backward compatibility, the source parameter that is described below is still supported, although its use is not recommended.

Submit this command in SAS to set up the connection for batch:

$r3connb (source = libref.catalog.entry SOURCE );

If you are using batch mode, submit this command in SAS to disconnect before exiting:

$r3connb (source = libref.catalog.entry SOURCE, function = close);

Passwords

In batch processing, all parameters to log on to an SAP system are generally stored in permanent SAS catalog entries. If you do not want to store all parameters—particularly, user ID, password, and client—use the following SAS code to prompt
the user for this information. In this scenario, you must enter all other connection information—the function module name, the gateway host, or the gateway service—in the Application Setup window when you create the SAS DATA step view.

/*----------------------------------------------------*/
/* This SAS program demonstrates how to prompt a user */
/* for the SAP user ID, password, client, and */
/* language information. */
/* The remote processing is enclosed in comments. */
/*----------------------------------------------------*/

*--- reset password ---*;
%let pwd=;

*--- set the message text ---*;
%let message=Please enter the SAP logon parameter;

*--- prompt the user for the parameters ---*;
%macro secure;
%global usr pwd cli lng;
%window R3 columns=80 rows=15
  #2 @5 message 50 protect=yes
  #4 @5 "User    : " usr 10 required=yes
  #6 @5 "Password: " pwd 10 display=no required=yes
  #8 @5 "Client   : " cli 3 required=yes
  #10 @5 "Language : " lng 1 required=yes;
%display R3;
%mend secure;

%secure;

/* === for remote processing ======================= */

*--- macro to pass macro variables to ---*;
*--- a remote SAS session ---*;
%macro syslput(macvar,macval,remote=);
%let str=%str(rsubmit &remote ;);
%nrstr(%let) %str(&macvar = &macval ; endrsubmit;);
&str;
%mend syslput;

*--- pass macro variables to the remote SAS session ---*;
%syslput(usr,&usr);
%syslput(pwd,&pwd);
%syslput(cli,&cli);
%syslput(lng,&lng);

*--- submit the code to the remote SAS session ---*;
 rsubmit;

=== end of remote processing =========================== */

*--- a temporary catalog entry for the connection ---*;
*--- parameters ---*;
%let source=work.r3conn.conn1.source;

*--- the connection id ---*;
*--- ATTENTION: This has to be the same connection ID ---*
*--- as the one that was used to create the views.    ---*
%let cconn=conn1;

*--- the host/port where the SAS RFC Server program ---*
*--- is running, used only in z/OS operating environment ---*
%let tcphost=cafe7.eur.sas.com;
%let port=6999;

*--- set the host parameter ---*
%let hst=hostname;

*--- save the parameters to the temporary catalog ---*
*--- entry ---*
  proc display c=sap33sr3dbi.savconnb.scl;
  run;

*--- connect to the SAP system ---*
%r3connb(source=&source);

*--- use the Data Step view you have created with the ---*
*--- SAS/ACCESS Interface to R/3 ---*

*--- close the connection to the SAP system ---*
%r3connb(source=&source, function=close);

/* === for remote processing
   ===================================
   endrsubmit;
   === end of remote processing
   =================================== */

*--- end of sample program ---*;

Dictionary

%CALLRFC Autocall Macro

Calls a specified RFC-enabled function module.

Type: autocall macro

Requirement: Function modules that are called using this macro must be RFC-enabled, synchronous, and have no user interaction.

Data source: R/3

See: RFC_LOGON_INFO macro variable, "Using the R/3 BAPI Connector: Logon Window"
Syntax

%CALLRFC
(<RFC-enabled function module name>
   EXPORTING <parameter string> IMPORTING <parameter string>
   INTABLES <parameter string> OUTTABLES <parameter string>
   USING <parameter string>)

Optional Arguments

EXPORTING
   passes field values or structures to the specified function module. The
   EXPORTING parameters are declared as import parameters in the function
   interface. Here is how they are defined.
   p1=value1 ... pn=valuen
      These parameters are passed as field values to the specified function
      module.
   p1.n1=value p1.n2=value2 ... pn.vn=valuen
      These parameters are passed as elements of a structured parameter. The
      parameter is a complex parameter that contains fields.

IMPORTING
   passes field values or structures from the specified function module back to
   SAS. Here is how they are defined.
   p1=var1 ... pn=varn
      The p1 ... pn parameters are export parameters that are declared in the
      function interface.
      The var1 ... varn parameters are SAS macro variables that are used to pass
      the values from the SAP system back to SAS. The macro variables must be
      defined before the function call is submitted.
   p1=dataset1 pn=datasetn
      Parameters that are written into a data set require a two-level data set name.
      Writing parameters into data sets can be useful for structured parameters.

INTABLES
   passes references to input SAS data sets. Here is how they are defined.
   p1=dataset1 ... pn=datasetn
      The SAS data sets are converted into internal tables and are passed to the
      specified function module. The variable names in the SAS data set must
      match the field names of the internal table as they are defined in the function
      interface.

OUTTABLES
   specifies references to output SAS data sets. Here is how they are defined.
   p1=dataset1 ... pn=datasetn
      Internal tables that are passed back to SAS from the specified function
      module are converted into SAS data sets.

USING
   specifies alternate connection parameters when you log on to the SAP system.
   By default, the %CALLRFC macro uses the connection parameters that are
   defined in the RFC_LOGON_INFO variable. However, if you want to use an
alternate set of connection parameters, create a variable that contains those parameters and reference that variable in the USING parameter. The alternate logon variable must be defined before the function call is submitted.

Details

If an error occurs while the %CALLRFC macro is executing, the macro variables &SYSRC, &SYSERR, or both contain a nonzero value. (See "Example 4: Check for Errors Using Macro Variables" on page 31.) The ABAP function module might also contain parameters to return error conditions to the calling program. These return codes can be written into a SAS macro variable and checked in the SAS program. (See "Example 5: Check for Errors Using Function Module Parameters" on page 32.)

Use the RFC_LOGON_INFO macro variable with the %CALLRFC macro. It defines the R/3 BAPI connection parameters that are required to access the SAP system. You can specify these connection parameters by using the RFC_LOGON_INFO macro string, or you can use the R/3 BAPI Connector window. This window is displayed automatically if you do not use the RFC_LOGON_INFO macro string when you submit your %CALLRFC macro statement. Below are the required connection parameters. If any required values are missing, you are prompted to provide the missing information.

- CLIENT
- USER
- PASSWD (or PASSWDX)
- LANG

The RFC_LOGON_INFO macro variable also requires some connection information. Here is how to define the connection parameters:

<table>
<thead>
<tr>
<th>If you use...</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td>the saprfc.ini file</td>
<td>DEST value</td>
</tr>
<tr>
<td>a specific application server</td>
<td>ASHOST and SYSNR values</td>
</tr>
<tr>
<td>load-balancing</td>
<td>MSHOST, R3NAME, and GROUP values</td>
</tr>
</tbody>
</table>

Load-balancing is available only if the CALLRFC executable has been linked using Release 4.0 or higher of the RFCSDK.

For more information about defining the RFC_LOGON_INFO variable, see "RFC_LOGON_INFO Automatic Macro Variable" on page 37. For more information about using the R/3 BAPI Connector window, see "Using the R/3 BAPI Connector: Logon Window" on page 24.
Examples:

Example 1: Read Data into a SAS Data Set

This example shows how to use the %CALLRFC macro to read data from an SAP R/3 system into a SAS data set. The %CALLRFC macro calls the RFC-enabled function module named RFC_SYSTEM_INFO and writes system information output to the WORK.RFC_SI_EXPORT SAS data set.

```sas
%callrfc(RFC_SYSTEM_INFO
   IMPORTING RFCSI_EXPORT=WORK.RFC_SI_EXPORT);
```

Example 2: Specify Logon Parameters

This example shows how to use RFC_LOGON_INFO to specify logon parameters for the %CALLRFC macro. TestUser connects to a specific application server to read data from the R/3 system into a temporary SAS data set named WORK.RFC_SI_EXPORT.

```sas
%let RFC_LOGON_INFO CLIENT=010 USER=TestUser
   PASSWD=TestPwd LANG=E
   ASHOST=HostName SYSNR=02;
%callrfc(RFC_SYSTEM_INFO
   IMPORTING RFCSI_EXPORT=WORK.RFC_SI_EXPORT);
```

Example 3: Use SAS Dates

This example shows how to use SAS dates in the function call. The %CALLRFC macro is used to call the function module named BAPI_COSTCENTER_GETLIST. The function call reads the list of cost centers for controlling area 1000 for the current date and writes the output list into a temporary SAS data set named WORK.COSTCENTER_LIST. The return code of the function call is written to the SAS macro variable MRETURN. The macro variable is defined before the %CALLRFC macro is used.

```sas
%let MRETURN=;
%callrfc(BAPI_COSTCENTER_GETLIST
   EXPORTING CONTROLLINGAREA=1000
      DATE=%sysfunc(date (), yymmddn8.)
   IMPORTING RETURN=MRETURN
   OUTTABLES COSTCENTER_LIST=WORK.COSTCENTER_LIST);
```

Example 4: Check for Errors Using Macro Variables

This example shows how to use the &SYSRC and &SYSERR macro variables to check for errors that occur while the %CALLRFC macro executes. A new macro named %EXAMPLE4 is created. It uses the SAS macro variables &SYSRC and &SYSERR to check return codes from the %CALLRFC macro. It also generates an error message if errors occurred.

```sas
%macro example4;
%callrfc(INVALID_FUNCTION_CALL);
  %if %eval(&SYSRC) ne 0 or %eval(&SYSERR) ne 0 %then %do;
    %put An error occurred while calling the function.
    %put SYSRC=&SYSRC SYSERR=&SYSERR;
  %end;
```

```sas
%mend example4;
```
Example 5: Check for Errors Using Function Module Parameters

This example shows how to use the parameters in the function module to check for errors that occurred while the %CALLRFC macro executes. A new macro named %EXAMPLE5 is created to retrieve a list of customers from the R/3 system. The range of customer numbers to retrieve is specified in the SAS data set WORK.IDRANGE. This data set is then used as the input table in the function call. Information for customers between customer number 0000000000 and 9999999999 is read into the SAS data set WORK.ADDRESSES. The function parameters contain return codes and messages from the BAPI_CUSTOMER_GETLIST function. The structure of the return parameter is defined in the function interface. In this example, the first character in the return string contains the message type. E indicates an error message and W indicates a warning message.

`%macro example5;
  %global bapi_return;
  /* create the input data set */
  data WORK.IDRANGE;
    sign='I';
    option='BT';
    low='0000000000';
    high='9999999999';
    output;
  run;
  %callrfc(BAPI_CUSTOMER_GETLIST
    IMPORTING RETURN=BAPI_RETURN
    INTABLES IDRANGE=WORK.IDRANGE
    OUTTABLES ADDRESSDATA=WORK.ADDRESSES);
  %if %substr(&bapi_return,1,1)=E or
    %substr(&bapi_return,1,1)=W %then %do;
    %put An error occurred while calling the BAPI_CUSTOMER_GETLIST function.;
    %put bapi_return=&bapi_return;
  %end;
  %else %do;
    proc print data=WORK.ADDRESSES;
    run;
  %end;
%mend;
%example5;`
Syntax

%R3CONNB
   (PROFILE=value, LIBREF=value, FUNCTION=open | close, ID=value, SOURCE=value)

Optional Arguments

PROFILE
   identifies the name of the profile that contains the information used in the Logon to R/3 window.

LIBREF
   identifies the SAS library containing the LIBREF.R3CONN data set. This data set contains the logon parameters defined for the profile.

FUNCTION
   specifies whether you want to open or close the connection to the R/3 system. Valid values are
   - OPEN
   - CLOSE

ID
   specifies the connection ID that is associated with the profile.

Note: This parameter was defined for previous versions of this macro and is supported in this release for backward compatibility. However, it is recommended that this parameter not be used.

SOURCE
   specifies the logon parameters that are defined in SOURCE type catalog entries. Although the SOURCE parameter is supported in this version, logon parameters should now be defined in profiles that are stored in a SAS data set.

Note: This parameter was defined for previous versions of this macro and is supported in this release for backward compatibility. However, it is recommended that this parameter not be used.

Details

The %R3CONNB macro lets you log on and off the SAP system using parameters stored in the data set libref.R3CONN under the profile name. Profiles that connect to and disconnect from the SAP system can be used for batch operation, such as overnight processing, and to simplify the logon process for end users.

Examples:

Example 1: Connect to an R/3 System By Using a Profile

The following example shows how to use the %R3CONNB macro to connect to the R/3 system using a profile:

%r3connb(PROFILE=BatchUser, LIBREF=batch, FUNCTION=open);
In this example, the profile BatchUser is used to connect to the R/3 system. The BatchUser profile uses the connection parameters stored in the BATCH.R3CONN data set.

Example 2: Disconnect from an R/3 System By Using a Profile

The following example shows how to use the %R3CONNB macro to disconnect from the R/3 system by using a profile:

```
%r3connb (PROFILE=BatchUser, LIBREF=batch, FUNCTION=close);
```

The BatchUser profile was used to connect to the R/3 system. This example demonstrates how to use the %R3CONNB macro to close the BatchUser profile’s connection to the R/3 system.

%R3CONNC Autocall Macro

Connects to an SAP system.

Type: batch macro
Data source: R/3
See: %R3CONNB macro, %R3CONNE macro on page 37

Syntax

```
%R3CONNC
    (<CCONN=value,> <HOST=value,> <PORT=value,> <USR=value,>
     <PWD=value,> <CLI=value,> <LNG=value,> <HST=value,>
     <DST=value,> <SNA=value,> <R3=value,> <SYS=value,>
     <GWS=value,> <GWH=value,> <FUNC=value,> <CPICUSR=value,>
     <CPICPWD=value,> <CPICCLI=value,> <CPICLNG=value,>
     <CPICDST=value,>
     <CPICFRM=value,> <DEBUG=value,> <REMSESS=value>)
```

Optional Arguments

**CCONN**
specifies the connection identifier to be used when connecting to the SAP system.

**HOST**
specifies the name of the TCP/IP host to be used when connecting to the SAS RFC Server. This parameter is used in z/OS operating environments only.

**PORT**
specifies the TCP/IP port to be used when connecting to the SAS RFC Server. This parameter is used in z/OS operating environments only.

**USR**
specifies the user ID to be used when connecting to the SAP system.

**PWD**
specifies the encoded password to be used when connecting to the SAP system.
CLI
specifies the three-digit client ID to be used with the SAP system.

LNG
specifies the logon language to be used with the SAP system.

HST
specifies the SAP server host name to be used when connecting to the SAP system.

DST
specifies the destination name to be used when using the SIDEINFO file to specify connection parameters.

SNA
specifies whether SAS connects to an R/2 system. Valid values are
- Y—SAS connects to an R/2 system.
- blank—SAS is not connecting to an R/2 system.

R3
specifies whether SAS is connecting to an R/3 system. Valid values are
- Y—SAS is connecting to an R/3 system.
- blank—SAS is not connecting to an R/3 system.

SYS
specifies the SAP System number. This parameter can be specified here or in the SIDEINFO file.

GWS
specifies the TCP service of the SAP gateway. This parameter can be specified here or in the SIDEINFO file.

GWH
specifies the name of the host on which the SAP gateway is running. This parameter can be specified here or in the SIDEINFO file.

FUNC
specifies the function module to be used.

CPICUSR
specifies the user identifier for CPIC.

CPICPWD
specifies the encoded password for CPIC.

CPICCLI
specifies the client for CPIC.

CPICLNG
specifies the logon language for CPIC.

CPICDST
specifies the destination for CPIC communication (as defined in the TXCOM table in the SAP system).

CPICFRM
specifies the form for CPIC.

DEBUG
specifies whether you want to use the trace option. Valid values are
- Y—Use the trace option.
REMSESS
specifies the SAS/CONNECT remote session ID to be used when connecting to the R/3 system.

Details
The %R3CONNC macro lets you connect to an SAP system. This macro can be used to log on to the SAP system during batch operation.

Note: The SIDEINFO file provides the connection parameters required for SAS to communicate with the SAP system. The SIDEINFO file is not needed if you specify all connection parameters in the %RCONNC macro.

Examples:

Example 1: Connect to a Specific R/3 System
The following example shows how to use the %R3CONNC macro to connect to an R/3 system when you specify a specific application server:

```sas
%r3connc(CCONN=IDES, USR=USER1, PWD=D27A927AD9E768,
          CLI=800, LNG=EN, HST=HostName1, SYS=02,
          CPICUSR=CPICUSER, CPICPWD=6FD3E4BAC4);
```

In this example, USER1 is connecting to an R/3 system on an application server named HostName1 with a system number of 02.

Example 2: Connect to an R/3 System Using an SAS RFC Server
The following example shows how to use the %R3CONNC macro to connect to an R/3 system and an SAS RFC Server that runs on a specified application server:

```sas
%r3connc(CCONN=IDES, USR=USER1, PWD=D27A927AD9E768, CLI=800, LNG=EN,
          HOST=HostName2, PORT=6998,
          HST=HostName1, SYS=02,
          CPICUSR=CPICUSER, CPICPWD=6FD3E4BAC4);
```

In this example, SAS connects to an R/3 application server (HostName1) with system number 02 and a SAS RFC Server running on a separate host (HOSTNAME2) on port 6998.

Example 3: Connect to an R/3 System Identified by a Logical Destination
This example shows how to use the %R3CONNC macro and a SIDEINFO file to connect to an R/3 system that is identified by a logical destination:

```sas
%r3connc(CCONN=IDES, USR=USER1, PWD=D27A927AD9E768,
          CLI=800, LNG=EN, DST=SYSTEM1,
          CPICUSR=CPICUSER, CPICPWD=6FD3E4BAC4);
```

In this example, USER1 is connecting to an R/3 system. The R/3 system is identified by a logical destination (SYSTEM1), and the connection information is defined in the SIDEINFO file.
%R3CONNE Autocall Macro

Disconnects from an SAP system.

Type: batch macro
Data source: R/3
See: %R3CONNB macro, %R3CONNC macro

Syntax

%R3CONNE(CCONN=value, REMSESS=value)

Optional Arguments

CCONN
specifies the connection identifier that is associated with the current session.

REMSESS
specifies the SAS/CONNECT remote session identifier that is associated with the current remote session.

Details

The %R3CONNE macro lets you disconnect from the SAP system. You can use this macro to log off from the SAP system during batch operation.

Example: Disconnect from an R/3 System

The following example shows how to use the %R3CONNE macro to disconnect from an R/3 system:

`%r3conne(CCONN=IDES);`

In this example, the %R3CONNE macro is used to terminate the R/3 connection with an ID of IDES.

RFC_LOGON_INFO Automatic Macro Variable

Defines the connection parameters that are required to access the SAP system.

Type: optional macro variable
Applies to: %CALLRFC macro
Data source: R/3
See: %CALLRFC macro, “Using the R/3 BAPI Connector: Logon Window”
Syntax

```plaintext
%let RFC_LOGON_INFO=ID=value ID=value ... ID=value
```

Details

RFC_LOGON_INFO is a macro variable that is used with the %CALLRFC macro. The RFC_LOGON_INFO variable uses the following parameters:

**Note:** Values are case sensitive. IDs are not.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>identifies the type of SAP system. Valid values are 3—(default) identifies an R/3 server E—identifies an external server.</td>
</tr>
<tr>
<td>CLIENT</td>
<td>is a required value that identifies the logon client.</td>
</tr>
<tr>
<td>USER</td>
<td>is a required value that specifies the user logon ID.</td>
</tr>
<tr>
<td>PASSWD</td>
<td>is a required value that identifies the logon password. This value is not required if a PASSWDX value is specified.</td>
</tr>
<tr>
<td>LANG</td>
<td>is a required value that identifies the logon language. Specify the one of the following: 1-byte SAP language (E for English, D for German, and so on) 2-byte ISO language (EN for English, DE for German, and so on)</td>
</tr>
<tr>
<td>LCHECK</td>
<td>specifies the logon check option at OPEN time. Valid values are 0—logon without check 1—(default) logon with check.</td>
</tr>
<tr>
<td>TRACE</td>
<td>specifies the RFC trace option. Valid values are 0—(default) without trace 1—with trace.</td>
</tr>
<tr>
<td>DEST</td>
<td>identifies the logical destination in the saprfc.ini file, if applicable. This value is required when you use the saprfc.ini file.</td>
</tr>
<tr>
<td>GWHOST</td>
<td>specifies the host name of the SAP gateway.</td>
</tr>
<tr>
<td>GWSERV</td>
<td>identifies the service of the SAP gateway.</td>
</tr>
<tr>
<td>Parameter ID</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MSHOST</td>
<td>specifies the host name of the message server if you use Load Balancing. This value is required if you use Load Balancing.</td>
</tr>
<tr>
<td>R3NAME</td>
<td>specifies the name of the SAP system if you use Load Balancing. This value is required if you use Load Balancing.</td>
</tr>
<tr>
<td>GROUP</td>
<td>specifies the name of the group of application servers if you use Load Balancing. This value is required if you use Load Balancing.</td>
</tr>
<tr>
<td>ASHOST</td>
<td>identifies the host name of the specific application server to be used. This value is required if you use a specific application server.</td>
</tr>
<tr>
<td>SYSNR</td>
<td>identifies the SAP system number if you use a specific application server and do not use Load Balancing. This value is required if you use a specific application server.</td>
</tr>
<tr>
<td>ABAP_DEBUG</td>
<td>specifies the ABAP debugger option. Valid values are 0—(default) run without the ABAP debugger 1—run with the ABAP debugger. Note: If you use the ABAP_DEBUG option, the SAP GUI must be installed.</td>
</tr>
<tr>
<td>PASSWDX</td>
<td>identifies the SAP logon password that is encrypted by SAS.</td>
</tr>
</tbody>
</table>

Examples:

**Example 1: Log In to a Server That Uses Load Balancing**

This example shows how to specify logon information for the `%CALLRFC macro using the RFC_LOGON_INFO macro variable:

```%
%let RFC_LOGON_INFO=CLIENT=010 USER=USER1
     PASSWD=USERPWD LANG=E
     MSHOST=HostName R3NAME=BIN GROUP=Public;
%callrfc(RFC_SYSTEM_INFO
     IMPORTING RFCSI_EXPORT=WORK.RFCSI_EXPORT);
```

In this example, USER1 is connecting to a server that uses Load Balancing to read R/3 data into a temporary data set called WORK.RFCSI_EXPORT.

**Example 2: Log In to a Specific Application Server**

This example shows how to specify logon information for the `%CALLRFC macro using the RFC_LOGON_INFO macro variable:

```%
%let RFC_LOGON_INFO=CLIENT=010 USER=USER1
```
PASSWD=USERPWD  LANG=E  
ASHOST=HostName  SYSNR=02;
%callrfc(RFC_SYSTEM_INFO
    IMPORTING RFCSI_EXPORT=WORK.RFCSI_EXPORT);

In this example, USER1 is connecting to a specific application server to read R/3 data into a temporary data set called WORK.RFCSI_EXPORT.
Using the LIBNAME Statement for SAP

Overview of the LIBNAME Engine for SAP ......................................................... 42
LIBNAME Statement Syntax for SAP ............................................................. 42
SAP LIBNAME Engine Options ................................................................. 43
SAS LIBNAME Statement Options for the SAP Engine ............................... 44
SAS Data Set Options for the SAP Engine .................................................. 45

Dictionary ........................................................................................................ 45
ABAPFM= LIBNAME Statement Option ......................................................... 45
ABAP_NAMESPACE= LIBNAME Statement Option ........................................... 45
ABAPPROG= LIBNAME Statement Option .................................................... 46
ASHOST= LIBNAME Statement Option ........................................................ 46
AUTHDOMAIN= LIBNAME Statement Option ................................................ 47
BATCH= LIBNAME Statement Option ........................................................... 47
BUFFER_SIZE= LIBNAME Statement Option ................................................ 48
CLASSIC LIBNAME Statement Option .......................................................... 48
CLIENT= LIBNAME Statement Option .......................................................... 49
DESTGROUP= LIBNAME Statement Option ..................................................... 49
DESTINATION= LIBNAME Statement Option ............................................... 49
GROUP= LIBNAME Statement Option .......................................................... 50
GWHOST= LIBNAME Statement Option ......................................................... 50
GWSERV= LIBNAME Statement Option ........................................................ 50
HOST= LIBNAME Statement Option ............................................................. 51
IEEE_REVERSE= LIBNAME Statement Option .............................................. 51
INENCODING= LIBNAME Statement Option ................................................ 52
LANGUAGE= LIBNAME Statement Option ..................................................... 52
MAX_TABLE_JOINS= LIBNAME Statement Option ........................................ 52
MODE= LIBNAME Statement Option ............................................................ 53
MSHOST= LIBNAME Statement Option .......................................................... 53
NAMESPACE= LIBNAME Statement Option .................................................... 53
NETWEAVER LIBNAME Statement Option ................................................... 54
NUMC_SAS_TYPE= LIBNAME Statement Option ........................................... 55
PASSWORD= LIBNAME Statement Option ..................................................... 55
PORT= LIBNAME Statement Option ............................................................. 56
R3NAME= LIBNAME Statement Option ........................................................ 56
RFC_STRING= LIBNAME Statement Option ................................................... 56
SAPLOGON_ID= LIBNAME Statement Option ................................................. 57
SCHEMA= LIBNAME Statement Option ......................................................... 57
SQL_FUNCTIONS_COPY= LIBNAME Statement Option ................................... 58
SYSNR= LIBNAME Statement Option ........................................................... 58
Overview of the LIBNAME Engine for SAP

The SAP LIBNAME engine is a read-only engine that lets you access tables and views from SAP systems. For most operating systems, the SAP LIBNAME engine makes a direct connection to the SAP system. For z/OS operating environments, the SAP LIBNAME engine accesses an SAP system through a SAS RFC Server. The SAP LIBNAME engine is part of the SAS solution to access data in an SAP system.

Note: Unlike other library engines, these interfaces are read-only and do not support any forms of data set creation, deletion, or modification.

LIBNAME Statement Syntax for SAP

```
LIBNAME libref R3 <options>;
```
**libref**
specifies a valid SAS name that serves as an alias to associate SAS with the SAP tables and view. It is any SAS name when you are assigning a new libref.

**R3**
explicitly specifies the R/3 library engine.

**options**
names one or more options honored by the R/3 engine, delimited with blanks.

The SAP LIBNAME statement assigns a library and allows the SAP engine to establish the connection to the SAP system. The SAP system checks the logon information submitted in the LIBNAME statement. The checked information includes the values for user, password, language, and client.

A user specifies a LIBNAME statement for the engine. It can then use that libref throughout SAS, wherever a libref is valid.

---

**SAP LIBNAME Engine Options**

These LIBNAME options apply to the SAP LIBNAME engine.

- "ABAPFM= LIBNAME Statement Option" on page 45
- "ABAP_NAMESPACE= LIBNAME Statement Option" on page 45
- "ABAPPROG= LIBNAME Statement Option" on page 46
- "ASHOST= LIBNAME Statement Option" on page 46
- "AUTHDOMAIN= LIBNAME Statement Option" on page 47
- "BATCH= LIBNAME Statement Option" on page 47
- "BUFFER_SIZE= LIBNAME Statement Option" on page 48
- "CLASSIC LIBNAME Statement Option" on page 48
- "CLIENT= LIBNAME Statement Option" on page 49
- "DESTGROUP= LIBNAME Statement Option" on page 49
- "DESTINATION= LIBNAME Statement Option" on page 49
- "GROUP= LIBNAME Statement Option" on page 50
- "GWHOST= LIBNAME Statement Option" on page 50
- "GWSERV= LIBNAME Statement Option" on page 50
- "HOST= LIBNAME Statement Option" on page 51
- "IEEE_REVERSE= LIBNAME Statement Option" on page 51
- "INENCODING= LIBNAME Statement Option" on page 52
- "LANGUAGE= LIBNAME Statement Option" on page 52
- "MAX_TABLE_JOINS= LIBNAME Statement Option" on page 52
- "MODE= LIBNAME Statement Option" on page 53
- "MSHOST= LIBNAME Statement Option" on page 53
- "NAMESPACE= LIBNAME Statement Option" on page 53
SAS LIBNAME Statement Options for the SAP Engine

The list below describes the SAP interface support for SAS /ACCESS LIBNAME options and presents default values where applicable. For more information, see *SAS/ACCESS for Relational Databases: Reference.*

- "CONNECTION= LIBNAME Statement Option" on page 60
- "CONNECTION_GROUP= LIBNAME Statement Option" on page 61
- "DBGEN_NAME= LIBNAME Statement Option" on page 61
- "DBPROMPT= LIBNAME Statement Option" on page 61
- "DBSASLABEL= LIBNAME Statement Option" on page 62
- "DEFER= LIBNAME Statement Option" on page 62
- "DIRECT_SQL= LIBNAME Statement Option" on page 62
- "MULTI_DATASRC_OPT= LIBNAME Statement Option" on page 63
- "REREAD_EXPOSURE= LIBNAME Statement Option" on page 63
- "SPOOL= LIBNAME Statement Option" on page 64
SAS Data Set Options for the SAP Engine

The list below describes the SAP interface support for SAS /ACCESS data set options and presents default values where applicable. For further information, refer to the SAS/ACCESS for Relational Databases: Reference.

- "BATCH= Data Set Option" on page 64
- "DBCONDITION= Data Set Option" on page 64
- "DBGEN_NAME= Data Set Option" on page 65
- "DBKEY= Data Set Option" on page 65
- "DBMASTER= Data Set Option" on page 65
- "NULLCHAR= Data Set Option" on page 66
- "NULLCHARVAL= Data Set Option" on page 66
- "REREAD_EXPOSURE= Data Set Option" on page 66

Dictionary

ABAPFM= LIBNAME Statement Option

Indicates the name of the ABAP function module that SAS uses.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: ABAPFUNC, ABAPFUNCTION
Default: /SAS/Z_SAS_DIALOG
Data source: R/3

Syntax

\texttt{ABAPFM= abap\_function\_name}

ABAP_NAMESPACE= LIBNAME Statement Option

Indicates the namespace for ABAP functions and programs that SAS uses.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: ABAPNAMESPACE, ABAP_NAME_SPACE, ABAPNS, ABAP_NS
Default: /SAS/
Data source: R/3

Syntax

**ABAP_NAMESPACE**= namespace

Details

If ABAP programs are installed in the customer namespace instead of in the /SAS/ namespace, this parameter identifies where the ABAP programs are installed.

**ABAPPROG= LIBNAME Statement Option**

Indicates the name of the ABAP program that SAS uses.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: ABAPPROGRAM, ABAPREPORT
Default: /SAS/Z_SAS_READ (the ABAP function module sets this)
Data source: R/3

Syntax

**ABAPPROG**= abap_program

**ASHOST= LIBNAME Statement Option**

Indicates the host name or IP address of a specific SAP application server.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: HST, R3HOST, RFHOST
Default: none
Data source: R/3

Syntax

**ASHOST**= application_server_host
AUTHDOMAIN= LIBNAME Statement Option

Indicates the name of the metadata-based authentication domain to use to associate user name and password credentials with an identity.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Default: none
Data source: R/3

Syntax

```
AUTHDOMAIN= authentication_domain
```

Details

In a metadata-based environment, credentials for an SAP system can be stored in a metadata repository. The user can reference the authentication domain in the LIBNAME statement so that the connection to the SAP system uses the credentials stored in metadata. This is used to provide user name and password information without using the USER= and PASSWORD= LIBNAME options.

BATCH= LIBNAME Statement Option

Indicates whether SAS should use SAP batch jobs for data extracts.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: BATCHMODE, BATCH_MODE
Default: N
Restriction: specific to SAP
Data source: R/3

Syntax

```
BATCH= 0 | 1 | Y | N
```

Required Arguments

- **Y | 1** specifies that SAS uses batch jobs to extract R/3 data.
- **N | 0** specifies that SAS uses dialog processes to extract R/3 data.
BUFFER_SIZE= LIBNAME Statement Option
Sets the minimum buffer size for data transfers in batch and dialog modes.
Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: BLOCKSIZE, BLOCK_SIZE, BUFFERSIZE, BUFFSIZE
Default: 100,000 bytes
Data source: R/3

Syntax
BUFFER_SIZE=buffersize

Details
The number of bytes must be greater than 10,000 and no more than eight digits.

CLASSIC LIBNAME Statement Option
Specifies that the engine uses the SAP classic RFC library.
Valid in: SAS/ACCESS LIBNAME statement
Default: none
Data source: R/3

Syntax
CLASSIC

Details
Note: In the third maintenance release for SAS 9.4, the CLASSIC LIBNAME option is available.

When you specify the CLASSIC option the engine uses the SAP classic RFC library.

If neither the CLASSIC or NETWEAVER option is specified in the LIBNAME statement or PROC CALLRFC, the ACCESS engine checks to verify whether the UNIX or Windows shared library is available. If the shared library can be loaded, the engine defaults to the SAP NetWeaver RFC libraries. If the shared library cannot be loaded, the engine defaults to the SAP classic RFC libraries.
CLIENT= LIBNAME Statement Option

Specifies the SAP logon parameter client.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: CLI, RFCCLIENT, RFCCLI
Default: SAP system default
Requirement: When you use the LIBNAME engine to access the SAP R/3 system or SAP BW system, you must specify valid logon information, including client, user name, password, and language.

Data source: R/3
Note: The SAP LIBNAME engine performs a logon check at OPEN time.
Example: 000 or 800

Syntax

CLIENT= client

DESTGROUP= LIBNAME Statement Option

Indicates the name of the destination group for batch access to the SAP system when the SAP LIBNAME engine uses direct RFC calls.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Default: SAS1
Data source: R/3

Syntax

DESTGROUP= destination_group

Details

Destination groups are defined in table /SAS/DESTS in the SAP system. Table /SAS/DESTS defines a group of RFC destinations.

DESTINATION= LIBNAME Statement Option

Indicates the destination in the saprfc.ini file if you are working with an saprfc.ini file.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Syntax

DESTINATION= destination

Details

If this is an R/3 system, you must also define the destination in the SIDEINFO file for the SAP gateway.

GROUP= LIBNAME Statement Option

Indicates the name of the group of SAP application servers if you are using load balancing.

Syntax

GROUP= application_server_group

GWHOST= LIBNAME Statement Option

Indicates the host name of the SAP gateway.

Syntax

GWHOST= gateway_host_name

GWSERV= LIBNAME Statement Option

Indicates the service of the SAP gateway.
**GWSERV= LIBNAME Statement Option**

Indicates the host of the SAS RFC Server to use to connect to the SAP system.

**Valid in:** SAS/ACCESS LIBNAME statement for SAP

**Alias:** GATEWAY_SERVICE

**Default:** none

**Data source:** R/3

**Syntax**

```
GWSERV= gateway_service
```
INENCODING= LIBNAME Statement Option

Indicates the code page.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Default: none
Data source: R/3
Note: Identifies characters and symbols that can be printed, displayed on terminals, and used in SAP programs.

Syntax

INENCODING= code_page

LANGUAGE= LIBNAME Statement Option

Indicates the SAP logon parameter language.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: LANG, LNG, RFCLANG, RFCLNG
Default: SAP system default
Requirement: When you use the SAP LIBNAME engine to access the SAP R/3 system or SAP BW system, you must specify valid logon information, including client, user name, password, and language.
Data source: R/3
Note: The SAP LIBNAME engine performs a logon check at OPEN time.
Example: EN, DE, or E, D

Syntax

LANGUAGE= language

Details
The value for language is either the 2-byte ISO-language key or the 1-byte SAP language.

MAX_TABLE_JOINS= LIBNAME Statement Option

Indicates the number of tables that you can use in a left-outer join or an inner join in ABAP Open SQL.
MODE= LIBNAME Statement Option
Indicates the mode for the SAP LIBNAME engine.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Default: DIRECT_SAP for operating environments other than z/OS. The default value for z/OS operating environments is RFC_SERVER.
Restriction: DIRECT_SAP and DIRECT values are supported on all platforms except z/OS. RFC_SERVER and SERVER values are supported only on z/OS.
Data source: R/3

Syntax

```
MODE= DIRECT_SAP | DIRECT | RFC_SERVER | SERVER
```
**NAMESPACE LIBNAME Option**

Specifies a string to subset a list of tables and views.

**Syntax**

\[ \text{NAMESPACE} = \text{name} \]

**Syntax Description**

- **name**: Specifies a string to subset a list of tables and views.

**Details**

The NAMESPACE= LIBNAME option is used to subset the list of tables and views that are returned by directory services. You will still be able to read other tables and views even if they are not listed.

---

**NETWEAVER LIBNAME Statement Option**

Specifies that the engine uses the SAP NetWeaver RFC library.

**Syntax**

\[ \text{NETWEAVER} | \text{NW} \]

**Details**

**Note:**

In the third maintenance release for SAS 9.4, the NETWEAVER LIBNAME option is available.

When you specify this option the engine uses the SAP NetWeaver RFC library.

If neither the NETWEAVER or CLASSIC option is specified in the LIBNAME statement or PROC CALLRFC, the ACCESS engine checks to verify whether the UNIX or Windows shared library is available. If the shared library can be loaded, the engine defaults to the SAP NetWeaver RFC libraries. If the shared library cannot be loaded, the engine defaults to the classic RFC libraries.

**Note:**

SAPLOGON_ID and TYPE are not supported by the NetWeaver RFC API. For more information about NetWeaver libraries, see your SAP documentation.
NUMC_SAS_TYPE= LIBNAME Statement Option

Indicates the SAS type for ABAP type NUMC.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: NUMC, NUMC_SASTYPE, NUMC_TYPE
Default: C
Data source: R/3

Syntax

NUMC_SAS_TYPE= N | C | $

Required Arguments

N
specifies a number.

C
specifies a character.

$ specifies a character.

PASSWORD= LIBNAME Statement Option

Indicates the SAP logon parameter password.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: PASS, PASSWD, PW, PWD
Default: none
Requirement: When you use the LIBNAME engine to access the SAP R/3 system or the SAP BW system, you must specify valid logon information, including client, user name, password, and language.
Data source: R/3
Note: The SAP LIBNAME engine performs a logon check at OPEN time.

Syntax

PASSWORD= password
PORT= LIBNAME Statement Option
Indicates the port number of the SAS RFC Server that is used to connect to the SAP system.
Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: PORT, RFC_SERVER_PORT, RFCSERVERPORT, SASRFC_SERVER_PORT, SASRFSERVERPORT
Default: none
Restriction: z/OS only
Data source: R/3

Syntax
PORT= rfc_server_port

R3NAME= LIBNAME Statement Option
Indicates the name of the R/3 system.
Valid in: SAS/ACCESS LIBNAME statement for SAP
Default: none
Data source: R/3

Syntax
R3NAME= system_name

RFC_STRING= LIBNAME Statement Option
Indicates additional logon or connection parameters for the RfcOpenEx() call.
Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: ADDITIONAL_RFC_OPTIONS, RFCOPENEX, RFCSTRING | RFC_OPTIONS_EXT
Default: none
Data source: R/3

Syntax
RFC_STRING= additional_rfc_options
Details
Connection parameters are used in the RfcOpenEx() call to log on to the SAP system. With this option, you can pass parameters that are not SAP LIBNAME options to the RfcOpenEx() call.

Example: RFC_STRING Example
RFC_STRING = "ABAP_DEBUG=1"

Note: When using RFC_STRING="ABAP_DEBUG=1", the ABAP debugger is invoked to debug the ABAP programs used by the engine. The SAP GUI on the SAS RFC Server host is required to use the ABAP_DEBUG=1 option.

SAPLOGON_ID= LIBNAME Statement Option
Indicates the string that is defined for SAPLOGON on 32-bit Windows.
Valid in: SAS/ACCESS LIBNAME statement for SAP
Default: none
Data source: R/3
Note: SAPLOGON_ID is not supported by the NetWeaver RFC API.

Syntax
SAPLOGON_ID= saplogon_id

SCHEMA= LIBNAME Statement Option
Subsets a table list by user name.
Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: OWNER
Default: none
Data source: R/3

Syntax
SCHEMA=schema-name

Syntax Description

schema-name
specifies a user ID to subset a list of tables and views.
Details
The SCHEMA= LIBNAME option is used to subset the list of tables and views that are returned by directory services. You will still be able to read other tables and views even if they are not listed.

SQL_FUNCTIONS_COPY= LIBNAME Statement Option
Writes the function associated with this particular LIBNAME statement to a table or the SAS log.
Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: none
Data source: R/3

Syntax
SQL_FUNCTIONS_COPY=libref.member | SASLOG

Syntax Description
libref.member
writes the current in-memory function list to a user-specified table for this particular LIBNAME statement.

SASLOG
writes the current in-memory function list to the SAS log for this particular LIBNAME statement.

SYSNR= LIBNAME Statement Option
Indicates the SAP system number—the 2-byte code that identifies the system on the host.
Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: SYS, SYSTEM, SYSNO
Default: none
Data source: R/3
Example: 00 or 01

Syntax
SYSNR= system_number
TRACE= LIBNAME Statement Option

Indicates whether SAS should trace the RFC requests.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Default: 0
Data source: R/3

Syntax

TRACE= 0 | 1 | Y | N

Required Arguments

Y | 1
specifies that RFC trace is switched on.

N | 0
specifies that RFC trace is switched off.

Details

If you switch on the trace option, SAS writes log information into a file. The RFC library logs messages in the dev_rfc file.

TRANSERRORCOMPATMODE= LIBNAME Statement Option

Indicates whether transcoding errors are reported.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Alias: TRANSERRORCOMPATMODE=, TRANSCODEERRORCOMPATMODE=,
TRANSCODING_ERROR_COMPATIBILITY_MODE=,
TRANS_ERROR_COMPATIBILITY_MODE=, TRANS_ERROR_COMPATMODE=
Default: N
Data source: R/3

Syntax

TRANSERRORCOMPATMODE=0 | 1 | Y | N
Required Arguments

0 | N
specifies that transcoding errors are reported either as a warning or an error, depending on the DATA step or procedure.

1 | Y
specifies that transcoding errors are ignored, as they were in releases prior to SAS 9.3.

Details
For a PROC SQL SELECT* FROM statement, only a warning is displayed when a transcoding error occurs.

---

**USER= LIBNAME Statement Option**

Indicates the SAP logon parameter user.

**Valid in:** SAS/ACCESS LIBNAME statement for SAP

**Alias:** RFCUSER, USERID, USERNAME, USR

**Default:** none

**Requirement:** When you use the LIBNAME engine to access the SAP system, you must specify valid logon information, including the client, user name, password, and language.

**Data source:** R/3

**Note:** The SAP system performs a logon check at OPEN time.

**Syntax**

```
USER= user
```

---

**CONNECTION= LIBNAME Statement Option**

Specifies whether operations on a single libref share a connection to the DBMS and whether operations on multiple librefs share a connection to the DBMS.

**Valid in:** SAS/ACCESS LIBNAME statement

**Default:** DBMS-specific. The default value for SAP is SHAREDREAD

**Data source:** R/3

**See:** CONNECTION= LIBNAME option in SAS/ACCESS for Relational Databases: Reference

**Syntax**

```
CONNECTION= SHAREDREAD | UNIQUE | SHARED | GLOBALREAD | GLOBAL
```
CONNECTION_GROUP= LIBNAME Statement Option

Causes operations on multiple librefs and on multiple pass-through facility CONNECT statements to share a connection to the DBMS.

Valid in: SAS/ACCESS LIBNAME statement
Default: none
Data source: R/3
See: CONNECTION_GROUP= LIBNAME option in SAS/ACCESS for Relational Databases: Reference

Syntax

CONNECTION_GROUP= connection-group-name

DBGEN_NAME= LIBNAME Statement Option

Specifies how SAS automatically renames DBMS columns (when they contain characters that SAS does not allow, such as $) to valid SAS variable names.

Valid in: SAS/ACCESS LIBNAME statement
Default: DBMS
Data source: R/3
See: DBGEN_NAME= LIBNAME option in SAS/ACCESS for Relational Databases: Reference

Syntax

DBGEN_NAME= DBMS | SAS

DBPROMPT= LIBNAME Statement Option

Specifies whether SAS displays a window that prompts the user to enter DBMS connection information before connecting to the DBMS in interactive mode.

Valid in: SAS/ACCESS LIBNAME statement
Default: NO
Data source: R/3
See: DBPROMPT= LIBNAME option in SAS/ACCESS for Relational Databases: Reference
Syntax

DBPROMPT= YES | NO

**DBSASLABEL= LIBNAME Statement Option**

Specifies the column labels an engine uses.

Valid in: SAS/ACCESS LIBNAME statement for SAP
Default: COMPAT
Data source: R/3
See: DBSASLABEL= LIBNAME option in SAS/ACCESS for Relational Databases: Reference

Syntax

DBSASLABEL= COMPAT | NONE

Required Arguments

COMPAT
specifies that the column labels are compatible with engine behavior in previous SAS releases. The SAP engine returns the short descriptive text for the columns of the SAP table. It reads the column label from the SAP data dictionary.

NONE
specifies that no column label information is returned.

**DEFER= LIBNAME Statement Option**

Specifies when the connection to the DBMS occurs.

Valid in: SAS/ACCESS LIBNAME statement
Default: NO
Data source: R/3
See: DEFER= LIBNAME option in SAS/ACCESS for Relational Databases: Reference

Syntax

DEFER= NO | YES

**DIRECT_SQL= LIBNAME Statement Option**

Lets you specify whether generated SQL is passed to the DBMS for processing.

Valid in: SAS/ACCESS LIBNAME statement
DIRECT_SQL= LIBNAME option in SAS/ACCESS for Relational Databases: Reference

Syntax

DIRECT_SQL= YES | NO | NONE | NOGENSEQNL | NOWHERE | NOFUNCTIONS | NOMULTOUTJOINS

MULTI_DATASRC_OPT= LIBNAME Statement Option

Used instead of the DBKEY= data set option to improve performance when you process a join between two data sources.

Valid in: SAS/ACCESS LIBNAME statement
Default: NONE
Data source: R/3
See: MULTI_DATASRC_OPT= LIBNAME option in SAS/ACCESS for Relational Databases: Reference

Syntax

MULTI_DATASRC_OPT= NONE | IN_CLAUSE

REREAD_EXPOSURE= LIBNAME Statement Option

Specifies whether the SAS/ACCESS engine behaves like a random-access engine for the scope of the LIBNAME statement.

Valid in: SAS/ACCESS LIBNAME statement
Default: NO
Data source: R/3
See: REREAD_EXPOSURE= LIBNAME option in SAS/ACCESS for Relational Databases: Reference.

Syntax

REREAD_EXPOSURE= NO | YES
**SPOOL= LIBNAME Statement Option**

Specifies whether SAS creates a utility spool file during read transactions that read data more than once.

Valid in: SAS/ACCESS LIBNAME statement

Default: YES

Data source: R/3

See: SPOOL= LIBNAME option in SAS/ACCESS for Relational Databases: Reference

**Syntax**

```
SPOOL= YES | NO
```

**BATCH= Data Set Option**

Specifies whether SAS uses SAP batch jobs for data extracts.

Valid in: DATA and PROC steps (when accessing DBMS data using SAS/ACCESS software)

Alias: BATCHMODE, BATCH_MODE

Default: N

Data source: R/3

Note: The BATCH= option is specific to SAP.

**Syntax**

```
BATCH= 0 | 1 | Y | N
```

**Required Arguments**

- Y | 1
  - SAS uses batch jobs to extract SAP data.

- N | 0
  - SAS uses dialog processes to extract SAP data.

**DBCONDITION= Data Set Option**

Specifies criteria for subsetting and ordering DBMS data.

Valid in: DATA and PROC steps (when accessing DBMS data using SAS/ACCESS software)

Default: none

Data source: R/3
See: 

DBCONDITION= data set option in SAS/ACCESS for Relational Databases: Reference.

Syntax

```plaintext
DBCONDITION= "DBMS-SQL-query-clause"
```

### DBGEN_NAME= Data Set Option

Specifies how SAS automatically renames columns (when they contain characters that SAS does not allow, such as $) to valid SAS variable names.

- **Valid in:** DATA and PROC steps (when accessing DBMS data using SAS/ACCESS software)
- **Default:** DBMS
- **Data source:** R/3
- **See:** DBGEN_NAME= data set option in SAS/ACCESS for Relational Databases: Reference

Syntax

```plaintext
DBGEN_NAME= DBMS | SAS
```

### DBKEY= Data Set Option

Specifies a key column to optimize DBMS retrieval.

- **Valid in:** DATA and PROC steps (when accessing DBMS data using SAS/ACCESS software)
- **Default:** none
- **Data source:** R/3
- **See:** DBKEY= data set option in SAS/ACCESS for Relational Databases: Reference

Syntax

```plaintext
DBKEY= ("<column-1"> <... <"column-n">)
```

### DBMASTER= Data Set Option

Designates which is the larger table when you process a join of tables from two different types of databases.

- **Valid in:** DATA and PROC steps (when accessing DBMS data using SAS/ACCESS software)
- **Default:** none
- **Data source:** R/3
NULLCHAR= Data Set Option

Indicates how missing SAS character values are handled during insert, update, DBINDEX=, and DBKEY= processing.

Valid in: DATA and PROC steps (when accessing DBMS data using SAS/ACCESS software)
Default: SAS
Data source: R/3
See: NULLCHAR= data set option in SAS/ACCESS for Relational Databases: Reference

Syntax

```plaintext
NULLCHAR= SAS | YES | NO
```

NULLCHARVAL= Data Set Option

Defines the character string that replaces missing SAS character values during insert, update, DBINDEX=, and DBKEY= processing.

Valid in: DATA and PROC steps (when accessing DBMS data using SAS/ACCESS software)
Default: a blank character
Data source: R/3
See: NULLCHARVAL= data set option in SAS/ACCESS for Relational Databases: Reference

Syntax

```plaintext
NULLCHARVAL= 'character-string'
```

REREAD_EXPOSURE= Data Set Option

Specifies whether the SAS/ACCESS engine behaves like a random-access engine for the scope of the LIBNAME statement.

Valid in: SAS/ACCESS LIBNAME statement
Default: NO
Data source: R/3
Examples:

Example 1: Accessing the SAP Server from a Local Host

In this example for the z/OS operating environment, the SAS RFC Server has been started on the local host on port 6998. You want to access the SAP system on sapr3srv.sup.com. The system number for the system is 03. Specify the entire set of user information, such as client, user, password, and language. If the SAS RFC Server was started on the SAS system host on port 6999, you do not need to define the host or port because 6999 is the default.

```r
libname mylib r3 user=TEST password=MYPASS client=800 language=EN
   ashost=sapr3srv.sup.com sysnr=03
   host=localhost port=6998;
```

Example 2: Changing the User Logon Language

You want to access the BW system on whjapp01 and the system number is 06. This time, you want to log on using the German language. If you log on using that language, the column labels are all in German.

```r
libname bwides r3 user=Test password=pwd client=800 language=DE
   ashost=whjapp01 sysnr=06;
```

Example 3: Defining an SAP Destination from the SAP Logon Utility

The SAP system that you want to access is defined as SI9 in the SAP logon utility.

```r
libname mylib r3 user=test password=secret client=800 language=E
   saplogon_id=SI9;
```

Example 4: Defining an SAP Destination from the saprfc.ini File

The SAP system that you want to access is defined by the bwides logical destination in the saprfc.ini file.

```r
libname mylib r3 user=test password=secret client=800 language=EN
   destination=bwides;
```

Example 5: Accessing the SAP Server with Credentials Stored in Metadata

User name and password credentials for the SAP system that you want to access are stored in metadata. The credentials are associated with an authentication domain named SAPAuth. The SAS Metadata Server is named meta.sup.com.

options metaserver=meta.sup.com metauser=test metapass=secret;
libname bwides r3 authdomain=SAPAuth client=800
  language=EN ashost=sapr3srv.sup.com sysnr=03;
CALLRFC Procedure

Overview: CALLRFC Procedure

The CALLRFC procedure executes Remote Function Calls (RFC) or RFC-compatible functions on an SAP system.

Any Advanced Business Applications Programming (ABAP) function modules that the CALLRFC procedure calls must be RFC-enabled, have no dialog boxes, and synchronous.

Syntax: CALLRFC Procedure

```
PROC CALLRFC <options>;
   CALL <FUNCTION> 'function-name'
      <EXPORTING parameter-1
         | parameter-1.field-1=value-1 < ... parameter-n=value-n>>
      <IMPORTING parameter-1=SAS-dataset-name-1
         < ... parameter-n=SAS-dataset-name-n>>
```

Examples: CALLRFC Procedure

Example 1: Displaying RFC_SYSTEM_INFO

Example 2: Using the Output Delivery System with the DESCRIBE FUNCTION Statement
PROC CALLRFC Statement

Invokes the CALLRFC procedure and specifies the connection and logon information of an SAP system. The connection to the SAP system is established and the logon information is verified.

Syntax

PROC CALLRFC <options>;

Summary of Optional Arguments

ABAPFM='abap-function-name'
specifies the name of the ABAP function module that SAS uses internally for metadata requests.

ABAPPROG='abap-program'
specifies the name of the ABAP program that SAS uses.

TRACE='RFC-trace-flag'
enables tracing of RFC requests.

Specify connection and logon parameters

CLASSIC
specifies that the engine and the CALLRFC procedure use the SAP classic RFC library.

LIBREF=SAP-engine-libref
Access the SAP system using connection and logon parameters associated with a libref.

NETWEAVER
specifies that the engine and the CALLRFC procedure use the SAP NetWeaver RFC library.
RFC_STRING='additional-RFC-options'
specifies additional logon or connection parameters for the RfcOpenEx() call.

Specify connection parameters
  _ID='SAP-logon-id'
specifies the string defined for SAPLOGON on 32-bit Windows.
  ASHOST='SAP-application-server-host'
specifies the host name or IP address of a specific application server.
  DESTINATION='SAP-destination'
Specify the destination defined in saprfc.ini to connect to the SAP system.
  GROUP='SAP-application-server-group'
specifies the name for a group of application servers.
  GWHOST='SAP-gateway-host-name'
specifies the host name of the SAP gateway.
  GWSERV='SAP-gateway-service'
  MSHOST='SAP-message-server-host'
specifies the host name of the Message Server for the SAP system.
  R3NAME='SAP-system-name'
specifies the name of the SAP R/3 system.
  SYSNR=SAP-system-number
specifies the SAP R/3 system number.

Specify data type and handling options
  IEEE_REVERSE=flag
specifies whether floating-point numbers are byte-reversed.
  NUMC_SAS_TYPE='N' | 'C' | '$'
specifies the SAS data type for the ABAP type NUMC.

Specify logon parameters
  CLIENT=SAP-client-number
specifies the SAP logon parameter for client.
  LANGUAGE=SAP-logon-language
specifies the SAP logon parameter for language.
  PASSWORD='SAP-password'
specifies the SAP logon parameter for password.
  PASSWORDX='encrypted-SAP-password'
specifies the SAP logon parameter for password in an encrypted form.
  USER=SAP-userID
specifies the SAP logon parameter for user.

Specify SAS RFC Server parameters (z/OS only)
  HOST='RFC-server-host'
specifies the host name of the SAS RFC Server.
  MODE=DIRECT_SAP | RFC_SERVER
specifies the mode for the CALLRFC procedure.
  PORT='RFC-server-port'
specifies the port number of the SAS RFC Server.
Optional Arguments

**ABAPFM=**`abap-function-name`

specifies the name of the ABAP function module that SAS uses internally for metadata requests.

Note: This option is not the name of the ABAP function that is called in the CALL FUNCTION statement.

Alias ABAPFUNCTION=, ABAPFUNC=

Default **Z_SAS_DIALOG**

**ABAPPROG=**`abap-program`

specifies the name of the ABAP program that SAS uses.

Alias ABAPREPORT=, ABAPPROGRAM=

Default **Z_SAS_READ** and is set by the ABAP function module

**ASHOST=**`SAP-application-server-host`

specifies the host name or IP address of a specific application server. This option is used for SAP R/3 when it is not configured for load balancing.

Alias HST=, RFCHOST=, R3HOST=

**CLASSIC**

specifies that the engine and the CALLRFC procedure use the SAP classic RFC library.

**CLIENT=**`SAP-client-number`

specifies the SAP logon parameter for client. Examples for a client are 000 or 800. This is a required SAP LIBNAME engine option. The SAP system performs a logon check at OPEN time.

Alias CLI=, RFCCLIENT=, RFCCLI=

Default The default value for this option is the SAP system default.

**DESTINATION=**`SAP-destination`

specifies the destination in saprfc.ini, if you are working with a saprfc.ini file to connect to the SAP system. If the SAP system is an R/3 system, you must also define this destination in the SIDEINFO file for the SAP gateway.

Alias DEST=, DST=, DSTN=

**GROUP=**`SAP-application-server-group`

specifies the name for the group of application servers when the SAP system is configured for load balancing.

**GWHOST=**`SAP-gateway-host-name`

specifies the host name of the SAP gateway.

Alias GATEWAY_HOST=

**GWSERV=**`SAP-gateway-service`

specifies the service of the SAP gateway.
**HOST=**`'RFC-server-host'`  
specifies the host name of the SAS RFC Server. This option is supported in z/OS operating environments only.

**IEEE_REVERSE=**`flag`  
specifies whether floating-point numbers are byte-reversed.

**LANGUAGE=** `SAP-logon-language`  
specifies the SAP logon parameter for language. The value for language is either the 2-byte ISO-language key or the 1-byte SAP language. Examples for the language are EN, DE or E, or D. The SAP system performs a logon check at OPEN time.

**LIBREF=** `SAP-engine-libref`  
specifies the libref for the library that is used to access the SAP system. The connection and logon information is copied from the libref. Because the connection and logon information is copied from the libref, this is the easiest way of using PROC CALLRFC.

**MODE=**`DIRECT_SAP | RFC_SERVER`  
specifies the mode for the SAP LIBNAME engine. If DIRECT_SAP is specified, the SAP LIBNAME engine uses RFC calls to the SAP system directly. If RFC_SERVER is specified, a SAS RFC Server is used to communicate with the SAP system.

**MSHOST=**`'SAP-message-server-host'`  
specifies the host name of the Message Server when the SAP system is configured for load balancing.

**NETWEAVER**  
specifies that the engine and the CALLRFC procedure use the SAP NetWeaver RFC library.

**Note**  
SAPLOGON_ID and TYPE are not supported by the NetWeaver RFC API. For more information about NetWeaver libraries, see your SAP documentation.
NUMC_SAS_TYPE='N' | 'C' | 'S'
specifies the SAS data type for the ABAP type NUMC. In SAS 8 generated views, it was N, but C is a better match. Setting NUMC_SAS_TYPE='N' ensures SAS 8 compatibility. The macro %R3CONNC sets this option to 'N' by default.

Alias NUMC_SASTYPE=, NUMC_TYPE=, NUMC=

PASSWORD='SAP-password'
specifies the SAP logon parameter for password. When using the SAP LIBNAME engine, either use the USER and PASSWORD options to specify logon credentials, or use the AUTHDOMAIN option to retrieve credentials from metadata. If single sign-on is used, this option is not needed. The SAP system performs a logon check at OPEN time.

Alias PASSWD=, PWD=, PW=, PASS=

PASSWORDX='encrypted-SAP-password'
specifies the SAP logon parameter for password in an encrypted form. It uses the encryption available with SAS/ACCESS Interface to R/3 for SAS 8 and is supported for compatibility with the SAS 8 connection profile.

Alias PASSWDX=, PWDX=, PWX=, PASSX=

PORT='RFC-server-port'
This option is supported in z/OS operating environments only. This option specifies the port number of the SAS RFC Server that is used to connect to the SAP system.

Alias RFC_SERVER_PORT=, RFCSERVERPORT=, SASR_SRV_PORT=, SASRFCSERVERPORT=

Default 6999

R3NAME='SAP-system-name'
specifies the name of the SAP R/3 system when the SAP system is configured for load balancing.

RFC_STRING='additional-RFC-options'
specifies additional logon or connection parameters for the RfcOpenEx() call. This call is used to log on the SAP system. Using this option enables passing parameters to the RfcOpenEx call that are not SAP LIBNAME engine options. This option can be used to support future extensions of the RfcOpenEx call.

Alias RFCSTRING=, RFC_OPTIONS_EXT=, RFCOPENEX=, ADDITIONAL_RFC_OPTIONS=

_ID='SAP-logon-id'
specifies the string defined for SAPLOGON on 32-bit Windows.

SYSNR=SAP-system-number
specifies the SAP R/3 system number. This option is used for SAP R/3 when it is not configured for load balancing.

Alias SYS=, SYSTEM=, SYSNO=

TRACE='RFC-trace-flag'
determines whether the SAP RFC library traces the RFC requests. The default directory location is read from the RFC_TRACE_DIR environment variable. The RFC library logs messages in the dev_rfc file in the same directory.
CALL FUNCTION Statement

Call an RFC on the SAP system. The CALL FUNCTION statement can be called multiple times within a PROC CALLRFC step. The function calls are executed in sequence.

Syntax

CALL <FUNCTION> 'function-name'
<EXPORTING parameter-1 | parameter-1.field-1=value-1
< ... parameter-n=value-n>>
<IMPORTING parameter-1=SAS-dataset-name-1
< ... parameter-n=SAS-dataset-name-n>>
<INTABLES table-parameter-1=SAS-dataset-name-1
< ... table-parameter-n=SAS-dataset-name-n>>
<TABLES table-parameter-1=SAS-dataset-name-1
< ... table-parameter-n=SAS-dataset-name-n>>
<CALLBACK <callback-parameter-1=SAS-dataset-name-1
< ... table-parameter-n=SAS-dataset-name-n>> <default = SAS-library>>;

Required Argument

function-name
specifies the name of the ABAP function. The function name is a literal, and it must be enclosed in quotation marks. The function name is converted to uppercase characters. The ABAP function must be RFC callable. It must not have any dialog, and it must be synchronous.

Optional Arguments

EXPORTING
specifies fields or structures to be passed to the function module. The parameters parameter-1 through parameter-n are defined in the function interface as import parameters.

IMPORTING
specifies the fields or structures to be passed from the function module back to the SAS system. The parameters parameter-1 through parameter-n are defined in the function interface as export parameters.
**INTABLES**

specifies the SAS data sets to be passed to the function module. The parameters table-parameter-1 through table-parameter-n are defined in the function interface as table parameters. SAS converts the SAS data sets into internal tables and passes it to the function module. The variable names in the SAS data set must match the field names of the internal table as defined for the function interface.

**TABLES**

specifies names for the SAS data sets to be created from the internal tables in the function module. The parameters table-parameter-1 through table-parameter-n are defined in the function interface as table parameters.

**CALLBACK**

specifies names for the SAS data sets to be created from the internal tables in the function module. The parameters are not defined in the function interface. The function module must implement a callback mechanism.

If the callback table name matches any of the names in the list of callback-parameters specified in the function call, the data is saved into the specified SAS data set. If the name does not match, but the default library is specified, then the data is written to `libref.table-name`.

---

**DESCRIBE FUNCTION Statement**

Reads metadata about an RFC and writes the information to the SAS log or to a SAS data set using ODS. The DESCRIBE FUNCTION statement can be called multiple times within a PROC CALLRFC step.

**Syntax**

```
DESCRIBE <FUNCTION> 'function-name' <SHORT>;
```

**Required Argument**

`function-name`

specifies the name of the ABAP function. The function name is a literal and it must be enclosed in quotation marks. The function name is converted to uppercase characters.

**Optional Argument**

`SHORT`

specifies that parameter metadata only is printed in the SAS log. The field metadata is not printed.

**Details**

**General Attributes**

In SAS listing output, the function name is printed. If ODS output is redirected to a SAS data set, here is the structure of the data set:
### Parameters

For each parameter, the following information is provided. If ODS output is redirected to SAS data sets, here is the structure of the data set:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
</table>
| class | Text 1 | Class | Parameter class:  
- I for input  
- O for output  
- T for table |
| name | Text 32 | Parameter | Parameter name |
| intlen | Number 8 | Internal length | Internal length in byte (ABAP) |
| leng | Number 8 | Length (number of characters) | |
| outlen | Number 8 | Output length | |
| decimals | Number 8 | Number of decimal places | |
| datatype | Text 5 | ABAP data type | |
| inttype | Text 1 | ABAP data type (internal) | |
| label | Text 256 | Label | |
### Name, Type, Label, Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Number 8</td>
<td>Variable Type</td>
<td>SAS variable type</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- 0 - no type, might be a structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- 1 - numeric</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- 2 - character</td>
</tr>
<tr>
<td>length</td>
<td>Number 8</td>
<td>Variable Length</td>
<td>SAS variable length</td>
</tr>
<tr>
<td>format</td>
<td>Text 32</td>
<td>Variable Format</td>
<td>SAS variable format</td>
</tr>
<tr>
<td>formatl</td>
<td>Number 8</td>
<td>Format Length</td>
<td>SAS variable format length</td>
</tr>
<tr>
<td>formatd</td>
<td>Number 8</td>
<td>Number of Format</td>
<td>SAS variable format decimals</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Decimals</td>
<td></td>
</tr>
<tr>
<td>informat</td>
<td>Text 32</td>
<td>Variable Informat</td>
<td>SAS variable informat</td>
</tr>
<tr>
<td>informl</td>
<td>Number 8</td>
<td>Informat Length</td>
<td>SAS variable informat length</td>
</tr>
<tr>
<td>informd</td>
<td>Number 8</td>
<td>Number of Informat</td>
<td>SAS variable informat decimals</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Decimals</td>
<td></td>
</tr>
</tbody>
</table>

### Columns

For each column of each structure (input, output, and tables), information in this table is listed. If ODS output is redirected to SAS data sets, here is the structure of the data set:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>parm</td>
<td>Text 32</td>
<td>Parameter</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>Text 32</td>
<td>Column</td>
<td></td>
</tr>
<tr>
<td>offset</td>
<td>Number 8</td>
<td>Offset</td>
<td></td>
</tr>
<tr>
<td>leng</td>
<td>Number 8</td>
<td>Length</td>
<td>(number of characters)</td>
</tr>
<tr>
<td>intlen</td>
<td>Number 8</td>
<td>Internal length</td>
<td></td>
</tr>
<tr>
<td>outlen</td>
<td>Number 8</td>
<td>Output length</td>
<td></td>
</tr>
<tr>
<td>decimals</td>
<td>Number 8</td>
<td>Number of decimal places</td>
<td></td>
</tr>
<tr>
<td>datatype</td>
<td>Text 5</td>
<td>ABAP data type</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Label</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>inttype</td>
<td>Text 1</td>
<td>ABAP data type (internal)</td>
<td></td>
</tr>
<tr>
<td>label</td>
<td>Text 256</td>
<td>Label</td>
<td>SAS variable type</td>
</tr>
<tr>
<td>type</td>
<td>Number 8</td>
<td>Variable Type</td>
<td>SAS variable type</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- 0 - no type, might be a structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- 1 - numeric</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- 2 - character</td>
</tr>
<tr>
<td>length</td>
<td>Number 8</td>
<td>Variable Length</td>
<td>SAS variable length</td>
</tr>
<tr>
<td>format</td>
<td>Text 32</td>
<td>Variable Format</td>
<td>SAS variable format</td>
</tr>
<tr>
<td>formatl</td>
<td>Number 8</td>
<td>Format Length</td>
<td>SAS variable format length</td>
</tr>
<tr>
<td>formatd</td>
<td>Number 8</td>
<td>Number of Format Decimals</td>
<td>SAS variable format decimals</td>
</tr>
<tr>
<td>informat</td>
<td>Text 32</td>
<td>Variable Informat</td>
<td>SAS variable informat</td>
</tr>
<tr>
<td>informl</td>
<td>Number 8</td>
<td>Informat Length</td>
<td>SAS variable informat length</td>
</tr>
<tr>
<td>informd</td>
<td>Number 8</td>
<td>Number of Informat Decimals</td>
<td>SAS variable informat decimals</td>
</tr>
</tbody>
</table>

**Usage: CALLRFC Procedure**

**SAP Classic RFC Libraries or SAP NetWeaver RFC Libraries**

In the third maintenance release for SAS 9.4, two new CALLRFC procedure options and LIBNAME options are available. Use the NETWEAVER option to specify that the engine uses the SAP NetWeaver RFC libraries. Use the CLASSIC option to specify that the engine uses the SAP classic RFC libraries. If neither option is specified in PROC CALLRFC or the LIBNAME statement, the ACCESS engine checks to verify whether the UNIX or Windows shared library is available. If the shared library can be loaded, the engine defaults to the SAP NetWeaver RFC libraries. If the shared library cannot be loaded, the engine defaults to the SAP classic RFC libraries.
Examples: CALLRFC Procedure

Example 1: Displaying RFC_SYSTEM_INFO

This example shows an example of retrieving the RFC_SYSTEM_INFO and displaying the information.

Program

``` Sas
proc callrfc ashost="sapr3srv.sup.com" sysnr='03' user='USER'
    passwd='PASS' client='800';
    call 'RFC_SYSTEM_INFO' importing rfcsi_export=work.rfcsi;
run;

proc transpose data=rfcsi out=rfcsi_transposed
    (drop=_name_ rename=(col2=Value _label_=Column));
    var _character_ _numeric_;
run;

proc print data=rfcsi_transposed;
run;
```

Output: Listing

<table>
<thead>
<tr>
<th>Obs</th>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RFC log version</td>
<td>011</td>
</tr>
<tr>
<td>2</td>
<td>Character set (SAP name)</td>
<td>1100</td>
</tr>
<tr>
<td>3</td>
<td>Integer format (1 / 2 = little / big endian)</td>
<td>LIT</td>
</tr>
<tr>
<td>4</td>
<td>Floating point format (1=IEEE, 2=IBM/370 format)</td>
<td>IE3</td>
</tr>
<tr>
<td>5</td>
<td>Logical destination (specified in function call)</td>
<td>sapr3srv_SRV_03</td>
</tr>
<tr>
<td>6</td>
<td>Character field of length 8</td>
<td>sapr3srv</td>
</tr>
<tr>
<td>7</td>
<td>R/3 system, name of R/3 system</td>
<td>SRV</td>
</tr>
<tr>
<td>8</td>
<td>R/3 system, name of R/3 system</td>
<td>SRV</td>
</tr>
<tr>
<td>9</td>
<td>Database host name</td>
<td>sapr3srv</td>
</tr>
<tr>
<td>10</td>
<td>R/3 system, name of central database system</td>
<td>MSSQL</td>
</tr>
<tr>
<td>11</td>
<td>R/3 system, system release</td>
<td>46C</td>
</tr>
<tr>
<td>12</td>
<td>RFC: SAP machine ID</td>
<td>560</td>
</tr>
<tr>
<td>13</td>
<td>R/3 system, operating system of application server</td>
<td>Windows NT</td>
</tr>
<tr>
<td>14</td>
<td>Time zone (difference from UTC in seconds)</td>
<td>-18000</td>
</tr>
<tr>
<td>15</td>
<td>Date and time, Daylight savings time flag</td>
<td>X</td>
</tr>
<tr>
<td>16</td>
<td>IP address</td>
<td>192.168.1.111</td>
</tr>
<tr>
<td>17</td>
<td>Kernel Release</td>
<td>46D</td>
</tr>
<tr>
<td>18</td>
<td>R/3 system, Name of application server</td>
<td>sapr3srv</td>
</tr>
<tr>
<td>19</td>
<td>Reserve field in RFCSI</td>
<td></td>
</tr>
</tbody>
</table>
Example 2: Using the Output Delivery System with the DESCRIBE FUNCTION Statement

This example shows using the Output Delivery System (ODS) to redirect the output for the DESCRIBE FUNCTION statement to SAS data sets.

The output data sets WORK.RFC_PARM, WORK.RFC_ATTR and WORK.RFC_COLS are created with the metadata information about function RFC_SYSTEM_INFO.

Program

```sas
ods listing close;
ods output parameter=rfc_parm attributes=rfc_attr column=rfc_cols;
proc callrfc user='USER' passwd='PASS' client='800'
    lang='EN' ashost='sapr3srv.sup.com' sysnr='03';
    describe function 'RFC_SYSTEM_INFO';
run;
ods output close;
```
SAP Basic Objects Supported by SAS/ACCESS Interface to R/3

SAS/ACCESS Interface to R/3 supports the following table types and views:
- transparent tables
- views
- pool tables
- cluster tables

SAP Basic Objects Supported by SAS Data Surveyor for SAP

SAS Data Surveyor for SAP supports the following table types and views:
- transparent tables
- views
- pool tables
- cluster tables

In addition, the SAS Data Surveyor for SAP supports reading the following basic InfoProviders from SAP BW:
- standard InfoCubes
- standard SAP HANA optimized InfoCubes
- real-time capable InfoCubes
- semantically partitioned InfoCubes
- standard DataStore objects
- write-optimized DataStore objects
- DataStore objects for direct update

Objects Supported in an SAP BW on an SAP HANA System

In an SAP BW on an SAP HANA system, SAP HANA views can be generated for SAP BW objects. Use the SAS/ACCESS Interface to SAP HANA to access those SAP HANA views. For more information, refer to *SAP HANA Modeling Guide* and other SAP documentation. For more information about SAS/ACCESS Interface to SAP HANA, refer to *SAS/ACCESS for Relational Databases: Reference.*