

Changes and Enhancements

Introduction

The following section describes the changes and enhancements introduced in Version 7 or Version 8 of SAS/CONNECT Software. Unless otherwise stated, these changes and enhancements apply to all of the operating environments that SAS/CONNECT supports.

Note: Information about changes and enhancements that were implemented in Version 8 is preceded by **V8**. All other changes and enhancements described in this section were implemented in Version 7. If your site upgraded directly from Version 6 to Version 8, then all of the changes and enhancements described here are new to you. If you upgraded from Version 7 to Version 8, then only the items preceded by **V8** are new to you. Δ

Asynchronous Remote Submits

SAS/CONNECT supports both synchronous and asynchronous remote processing. In Version 6, only synchronous processing was supported. Synchronous processing means that the processing on the remote host must be complete before you regain control of your SAS session. Asynchronous processing means that control is returned immediately so you can continue processing on the local host.

The following new statements are available for checking the status of asynchronous remote processing and obtaining results from it:

RDISPLAY

creates windows for the log and output from an asynchronous remote submit.

RGET

retrieves the log and output from an asynchronous remote submit and merges them with the local session.

New options have also been added to the SIGNON, the SIGNOFF and the RSUBMIT statements to support asynchronous processing. The following options help control access to and execution on the remote host:

CONNECTREMOTE=

identifies which remote session to connect to.

CONNECTSTATUS=

indicates whether the status window will be displayed during a file transfer.

CONNECTWAIT=

specifies whether a remote submit is processed synchronously or asynchronously.

CMACVAR=

identifies the name of a macro variable to associate with a remote submit.

CSCRIPT=

specifies a script file to use during SIGNON and SIGNOFF.

CSYSRPUTSYNC=

forces the macro variables for an asynchronous remote submit to be defined as soon as %SYSRPUT executes rather than at a synchronization point.

For information about SIGNON, SIGNOFF, and RSUBMIT, see “SIGNON Command and Statement” on page 227, “SIGNOFF Command and Statement” on page 233, and “RSUBMIT Command and RSUBMIT Statement” on page 21.

Output Delivery System (ODS) Support

When using compute services in SAS/CONNECT software, you can use the Output Delivery System (ODS) to change the format and appearance of SAS output that is generated on a remote host. You can generate ODS output from a remote host when using either a synchronous or an asynchronous remote submit. ODS statements and procedures that support ODS can be used to

- convert each piece of output into an "object" that can be manipulated and viewed
- specify options for different ODS destinations, such as HTML or a SAS data set
- create or select templates to customize the output
- select and exclude output from the ODS object.

For more information about ODS support, see “Output Delivery System” on page 18.

Data Encryption

SAS/CONNECT provides data encryption to guarantee the security of data that is sent across a network.

For the OS/390, OpenVMS, OS/2, and UNIX platforms, the purchase of a license for SAS/SECURE is necessary. SAS/SECURE provides access to the encryption services of the RSA BSAFE Crypto-C Toolkit.

▮ In Version 8, SAS/SECURE introduces support for the RSA BSAFE Crypto-J Toolkit, which supports Java clients that access SAS servers.

Encryption is specified by the following SAS options:

NETENCRYPTALGORITHM

specifies the encryption algorithm.

NETENCRYPT

specifies the requirement to operate in encryption mode; for example, both the client and the server sessions must be operating in encryption mode.

NETENCRYPTKEYLEN

specifies the key length, 40-bits or 128-bits, to be used by the encryption algorithm.

NETMAC

controls the use of Message Authentication Codes (MACs) on network communications. A MAC is the equivalent of a checksum that is used to ensure that the original message has not been modified.

For more information about data encryption, see Appendix 2, “Encryption Services,” on page 487.

Cross-Environment Data Access (CEDA)

SAS software introduces a new cross-environment data access (CEDA) facility. CEDA allows any SAS data file that is created on a directory-based host to be read by a SAS session that is running on another directory-based host. The platforms that CEDA supports are

- Windows NT
- Windows 95
- Windows 98
- OS/2
- Macintosh
- Solaris
- HP-UX
- RS/6000 AIX
- Compaq Tru64 UNIX (formerly Compaq’s DIGITAL UNIX)
- OpenVMS Alpha
- OpenVMS VAX
- OS/390 with UNIX Support Services

For more information about CEDA, see Chapter 13, “Using Cross-Environment Data Access (CEDA),” on page 93.

The UPLOAD and the DOWNLOAD Procedures

The following sections describe several enhancements that have been made to the UPLOAD and DOWNLOAD procedures. For more information about PROC UPLOAD and PROC DOWNLOAD, see Chapter 17, “The UPLOAD Procedure,” on page 107 and Chapter 18, “The DOWNLOAD Procedure,” on page 129.

Translation Technique

When both the local host and the remote host are running Version 7 or a later release of SAS/CONNECT software, the translation rules have changed. First, no translation occurs if both the remote and local hosts have the same machine representations. This prevents any unnecessary translation from occurring. If translation is needed, the receiving host translates the data directly into its native representation. So, when both the local and remote hosts are Version 7 or later, the data is translated only once when translation is necessary and is not translated when both hosts have the same machine architecture.

Using Wildcard Characters When Transferring Files

SAS/CONNECT supports the use of an asterisk (*) as a wildcard character for transferring external text files or external binary files. The wildcard character may be used with the INFILE= option to send

- all files in a specific library (*).
- all files with a specific extension (*.extension).
- all files with the same name, but different extensions (name.*).

Transferring SAS Files

The list of member types that can be transferred by using the MEMTYPE= option has been expanded to include:

- DATA (SAS data files)
- CATALOG | CAT (SAS catalogs)
- VIEWS (SQL views)
- MDDDB (multi-dimensional database)
- FDB (financial database)
- DMDB (data mining database)
- _ALL_ (all files in the library).

New Options

For PROC UPLOAD and PROC DOWNLOAD, you can use the following new options to help identify the files that will be transferred, the translation tables to use, and the transfer technique.

AFTER=

specifies that files are to be transferred only if they were modified on or after a specific date.

EXTENDSN=

specifies whether or not to promote the length of short numerics.

TRANIN=

specifies a translation table to be used when transferring a data set that is stored in a different data representation than that of the local host.

TRANNET=

specifies a translation table to be used when transferring a data set between local and remote hosts that have different machine architectures.

TRANOUT=

specifies a translation table to be used when you are transferring a data set, and the OUTREP= data set option indicates a data representation different from the native data representation of the target host.

V6TRANSPORT

specifies that data should be translated from a local format to a transport format. (The receiving host then translates the data from the transport format to a local format.)

Messaging Services

SAS/CONNECT offers the ability to design and develop distributed applications that communicate by using messages. Messaging services enables the programs that compose a distributed application to communicate by sending data in the form of a message. Two forms of messaging services are available: direct messaging and indirect messaging. Both direct and indirect messaging enable you to include attachments with messages that are sent between a client and a server portion of an application.

Direct Messaging

SCL programs can use direct messaging to communicate by sending messages directly between each other. However, both portions of the application must be running at the same time for the connection to be made.

For more information, see Chapter 29, “Using Direct Messaging,” on page 273.

Indirect Messaging

Indirect messaging enables programs to communicate indirectly by placing messages on queues. When you use message queues, the programs do not have to be running at the same time to communicate. SAS/CONNECT provides the following two interfaces for accessing message queues: an SCL interface, and a CALL routine interface, which is used with SAS DATA steps or SAS macros.

For more information, see Chapter 31, “Using Indirect Messaging,” on page 281.

Remote Objecting Services

SAS/CONNECT gives SAS/AF developers the ability to distribute selected portions of their encapsulated object frameworks across remote session boundaries.

For more information, see Chapter 20, “Using Remote Objecting Services,” on page 173.

DOMAIN Server

The DOMAIN server has been enhanced to provide queue management and agent scheduling in addition to protocol gateway services. A DOMAIN server is created by using PROC DOMAIN.

For more information, see Chapter 38, “The DOMAIN Server,” on page 411.

Queue Management

The queue manager controls the queues that are used by applications that employ the indirect-messaging facility.

For more information, see “Queue Management” on page 412.

Agent Scheduling Services

These services are used as a client/server-based implementation of a periodic job scheduler, with an extension to support on-demand execution.

For more information, see Chapter 40, “Using Agent Services,” on page 429 and “Agent Scheduling” on page 416.

Communications Access Methods

Information about communications access methods that was formerly in Version 6 *SAS/CONNECT Software: Usage and Reference* is now available in a new document, *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software*. This document supports SAS releases from Version 6 to Version 8 and consolidates all information about communications access methods from SAS/CONNECT documentation, SAS/SHARE documentation, and various technical reports.

Security Support Provider Interface

SAS software support the Microsoft security support provider interface (SSPI), which enables a Windows NT user to be transparently authenticated on another Windows NT machine. This enables SAS/CONNECT users to perform a scriptless signon to a spawner that resides on a remote Windows NT host without specifying a userid and a password. A remote SAS session will be created with the same security context as a local user. To find out more about this capability, see *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software*.

Support for General Enhancements

SAS/CONNECT supports these enhancements to base SAS software:

- long names
- concatenated libraries
- concatenated catalogs
- generations
- integrity constraints
- quoted strings
- SAS environment windows.

For more information about these features, see *SAS Language Reference: Dictionary* and *SAS Language Reference: Concepts*.

Version 8 Method to Secure the Remote Host

Version 8 offers a new method to secure a SAS/CONNECT remote host by means of the USER= and PASSWORD= options to the SAS/CONNECT RSUBMIT and SIGNON statements.

These security options can be set on any Version 8 SAS/CONNECT local host accessing a remote host that runs any version of SAS. The USER= and PASSWORD= options to these statements are recommended and take precedence over the applicable security option, which varies by host and access method. SAS/CONNECT security options are APPCSEC, APPC_SECURE, TCPSEC, and SASUSER and SASPASS.

To establish SAS/CONNECT security in Version 8, you specify the USER= and PASSWORD= options in the appropriate statement on the local host.

If a Version 7 security option remains set on the local host, the Version 8 specification of USER= and PASSWORD= in a SAS/CONNECT statement overrides the previously set security option on the local host. For example, the Version 8 USER= and PASSWORD= options in the SIGNON statement will override the TCPSEC=_PROMPT_ option set on a UNIX local host for a non-scripted sign on to a spawner.

If a Version 8 local host does not set USER= and PASSWORD= options, the communications access method or host security option would remain in effect. If both the USER= and PASSWORD= options and a security option are specified, then the USER= and PASSWORD= options would take precedence.

Syntax and definitions are:

USER | **USERNAME** | **USERID** | **UID=** *username* | **_PROMPT_**

PASSWORD | **PASSWD** | **PWD** | **PW=** *password* | **_PROMPT_**

Specifying these options allows local hosts whose usernames and passwords have been verified to access the remote host.

Username is a valid userid on the remote host that is being accessed. On Windows NT only, the username can also include the domain name, which locates the specified username in a domain.

Password is a valid password on the remote host that is being accessed.

Supplying a userid and password by using the USER= and PASSWORD= options is more secure than assigning them by means of a security option (such as TCPSEC), which can be inadvertently publicized in a configuration file or in a log .

PROMPT specifies that the SAS System prompts for *userid* and *password*.

Hardcoding a username and password value to the USER= and PASSWORD= options limits the assignment to a single user whereas prompting permits any user to supply a username and password that are valid. Specifying only USER=_PROMPT_ implies that the SAS System will prompt for both a username and a password.

The values supplied for the USER= and PASSWORD= options are valid for the duration of the remote host connection. Subsequent local host connections to the same remote host or to a different remote host require you to specify these options again. By contrast, as an example, the values assigned to TCPSEC in a local host configuration file endure for subsequent connections to the same remote host and to different remote hosts.

Here is a Version 8 example:

```
signon user=joeblack password=born2run;
```

As a security precaution, PASSWORD= field entries echoed in the local host log are replaced with Xs.

If **_PROMPT_** is specified, when presented with the prompt for *password* during a remote host connection, the value entered would not be displayed on the screen.

Version 8 %SYSLPUT Macro Statement

In Version 7, %SYSLPUT (used for creating a macro variable on the remote host) was released as a sample macro that could be downloaded from the SAS Online Samples.

∇8 In Version 8, %SYSLPUT has been implemented as a full macro statement. For details about the syntax, see “%SYSLPUT Statement” on page 32.

Version 8 Multi-Process (MP) CONNECT

∇8 Version 8 introduces support for parallel (or multi-) processing with Multi-Process (MP) CONNECT. This facility exploits a local host’s multi-processor capability by allowing parallel processing of self-contained tasks and the coordination of all the results into the original SAS session. SAS/CONNECT accomplishes multi-processing by means of a new SASCMD option, which establishes a connection to one or more “remote” SAS sessions that run on the local host. MP CONNECT’s “remote” session actually executes on the local host.

Note: MP CONNECT is available with the TCP/IP access method only. △

To use MP CONNECT, you specify options in the SIGNON and RSUBMIT statements in order to identify one or more tasks to be processed by a “remote” SAS session. The remote host identifier is supplied by the new PROCESS= option, which is an alias for the existing REMOTE= option. REMOTE= can also be used to specify the remote host identifier. In this context, the identifier does not identify the remote host but is an arbitrary name that you associate with a specific task.

MP CONNECT provides the SASCMD= option, which enables signon to the local host’s processor. The WAITFOR statement allows you to wait for one or more asynchronously executing tasks before returning control to the local session. The LISTTASK statement lists all active or completed tasks.

MP CONNECT introduces the PROCESS= option and the SASCMD= option that are used with the SIGNON and RSUBMIT statements:

PROCESS=remote-session-id

is an alias to the REMOTE= option. The PROCESS= option was added as an alias to allow you to differentiate between an MP CONNECT signon (PROCESS=) and a signon to a remote host on a network (REMOTE=). After the *remote-session-id* has been specified by either the SIGNON command or as a system option, subsequent RSUBMIT, SIGNON, or SIGNOFF commands or statements that omit *remote-session-id* default to the specified session id.

SASCMD=“SAS-command”

is the command that is used to invoke the remote SAS session on the local host. SASCMD is also a global SAS system option. If the SASCMD global system option is already set, a locally set SASCMD= option in the SIGNON command would take precedence over the globally set option.

Note: If you need to execute additional host commands prior to the SAS invocation, it is recommended that you write a host-specific script that contains your host commands and the SAS invocation, and specify this script as the SASCMD value. △

Version 8 Autosignon

∇8 Starting in Version 8, if an RSUBMIT command or statement is executed and no connection currently exists, RSUBMIT automatically executes a SIGNON and uses any global settings of SAS/CONNECT options. RSUBMIT connection options are also executed to establish a connection to the remote SAS session and then to execute the

RSUBMIT. Therefore, all of the valid SIGNON options are now valid for RSUBMIT. In addition, the default action is to automatically sign off at the end of the RSUBMIT.

Autosignon introduces the PERSIST= option in the RSUBMIT statement:

PERSIST = YES | NO

specifies whether a signoff is automatically performed from the remote session on the local host after the RSUBMIT has completed. RSUBMIT provides complete control over a remote session's termination.

Version 8 Asynchronous Task Statements

¶8 Two new statements enable you to monitor asynchronous tasks that are executed in an RSUBMIT command or statement.

WAITFOR	allows you to wait for any or all of a list of asynchronous tasks that are provided as arguments to this statement.
LISTTASK	lists the active tasks and the completed tasks that were instantiated by the current SAS session.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/CONNECT User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 537.

SAS/CONNECT User's Guide, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-477-2

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, September 1999

SAS[®] and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. [®] indicates USA registration.

IBM[®], AIX[®], DB2[®], OS/2[®], OS/390[®], RS/6000[®], System/370[™], and System/390[®] are registered trademarks or trademarks of International Business Machines Corporation. ORACLE[®] is a registered trademark or trademark of Oracle Corporation. [®] indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.