



## CHAPTER

## 14

## SAS/SHARE Macros

---

<i>Introduction</i>	133
<i>LIBDEF</i>	133
<i>Syntax</i>	134
<i>LISTLIB</i>	135
<i>Syntax</i>	135
<i>LISTSRV</i>	135
<i>Syntax</i>	135
<i>LISTSRVI</i>	135
<i>Syntax</i>	135
<i>OPERATE</i>	135
<i>Syntax</i>	136
<i>SERVERID</i>	136
<i>Syntax</i>	136
<i>SERVIIDX</i>	136
<i>Syntax</i>	136
<i>SERVINFO</i>	137
<i>Syntax</i>	137
<i>SERVLIB</i>	138
<i>Syntax</i>	138
<i>SETSRV</i>	139
<i>Syntax</i>	139
<i>SHRMACS</i>	139
<i>Syntax</i>	139
<i>SHUTSRV</i>	140
<i>Syntax</i>	140
<i>STRTSRV</i>	141
<i>Syntax</i>	141

---

### Introduction

The following sections describe each SAS/SHARE macro, show its syntax, and give examples of how it is used. For more details about how to use the macros, see Chapter 6, “Using SAS/SHARE Macros for Server Access,” on page 75.

---

### LIBDEF

The LIBDEF macro generates a LIBNAME statement.

## Syntax

```
%LIBDEF (libref,< SAS-data-library>, <READONLY>, <RETRY>,
  <ENGINE=>local-engine, <RENGINE=remote-engine>,
  <RMTVIEW=remote-engine-RMTVIEW=-option>, <SLIBREF=server-library>,
  <uapw>, <APPLSYS=appl-sys-lib-table>);
```

The LIBDEF macro generates a LIBNAME statement to define a SAS data library that will be accessed locally or through a server. The server administrator can specify a physical name or a reserved libref in the APPLSYS macro library. If the physical name is specified, the server administrator must supply it as the second argument to this macro; the physical name should not be enclosed in quotes. If a libref is specified, the administrator must omit the second argument.

If you specify a physical name that is not specified in the APPLSYS macro library, a LIBNAME statement is generated without the SERVER= option.

### READONLY

specifies that the server library be accessed in read-only mode.

### RETRY

If a LIBNAME statement specifying the SERVER= option fails, LIBDEF generates a LIBNAME statement without the SERVER= option.

### ENGINE=

specifies the local engine to be used in the user's session to access the library. The default is ENGINE=REMOTE. Omit this parameter unless you need to override the default engine.

### RENGINE=

specifies the remote engine to be used in the server session to access the library. There is no default for this parameter. SAS chooses an appropriate engine. Omit this parameter unless you need to override the engine that is chosen by SAS.

### RMTVIEW=

specifies the value of the REMOTE engine's RMTVIEW= option in the LIBNAME statement that the LIBDEF macro generates. This parameter can be used to override the default value of this option for a specific library. This parameter overrides the RMTVIEW= parameter to the SERVINFO and SERVLIB macros.

### SLIBREF=

specifies the libref that references the specified library in the server session. If this parameter is specified, the generated LIBNAME statement will include the SLIBREF= option. This parameter overrides any value that is specified for the *SAS-data-library* argument to the LIBDEF macro statement.

### *uapw*

specifies that a user can access the server only by supplying a password.

### APPLSYS=

specifies a new application system library table to be loaded. This argument is ignored if the table has already been specified in the SHRMACS call or in a previous LIBDEF call.

After executing the LIBDEF macro, the automatic macro variable SYSLIBRC contains the return code from the LIBNAME statement. See *SAS Guide to Macro Processing* for more information about automatic macro variables.

```
%libdef(mylib,SAS-data-library,applsys=qa); %libdef(perm,SAS-data-library,reado
%libdef(datalib);
```

---

## LISTLIB

Lists the current library-alias table.

---

### Syntax

```
%LISTLIB <FULL>;
```

The LISTLIB macro writes the library-alias table that is currently in use to the SAS log. It shows the server alias that is associated with each SAS data library.

**FULL**

optional argument that also writes the values that are specified for the SERVLIB macro parameters to the SAS log. For more information about using the SERVLIB macro parameters, see “Associating SAS Libraries with Server Aliases (SERVLIB)” on page 82.

```
%listlib full;
```

---

## LISTSRV

Lists the server-alias table.

---

### Syntax

```
%LISTSRV;
```

The LISTSRV macro writes the server-alias table to the SAS log. It shows the serverid that is associated with each defined alias.

```
%listsrv;
```

---

## LISTSRVI

Lists the server information table.

---

### Syntax

```
%LISTSRVI;
```

The LISTSRVI macro writes the server information table to the SAS log. It shows the REMOTE engine’s LIBNAME statement option RMTVIEW= and the network node name, by default.

```
%listsrvi;
```

---

## OPERATE

Generates a PROC OPERATE statement.

---

## Syntax

**%OPERATE**(*server-name*, <*oapw*>);

The OPERATE macro invokes the OPERATE procedure for a server that is identified by the *server-name* argument.

*server-name*

specifies the server name. It can be an alias or an actual serverid. This macro generates a single PROC OPERATE statement.

*oapw*

specifies the administrator password if one is required by the server. This value is mapped to the SAPW= option in the PROC OPERATE statement.

```
%operate(serv1,myoperpw);
```

---

## SERVERID

Converts a server alias to a serverid.

---

## Syntax

**%SERVERID**(*server-alias*, <NEQ>);

*server-alias*

The SERVERID macro converts the *server-alias* to an actual serverid. It supplies the actual serverid in the form SERVER= to the SERVER and OPERATE procedures and LIBNAME statements.

NEQ

supplies only the serverid value (without the SERVER= option).

Additionally, the SERVERID macro generates a %LET statement for a macro variable whose name is the high-level qualifier in a two-level server name in the following form:

```
%LET high-level-qualifier=network-node;
```

The server name must be listed in the server information table and have a network node name associated with it, as shown in these examples

```
libname mylib 'SAS-data-library' %serverid(devserv); set server %serverid(serv1
```

The first example generates the LIBNAME statement, which supplies the SERVER=*serverid* parameter. The second example generates a SET SERVER statement, which supplies only the *serverid* without the SERVER= parameter keyword.

---

## SERVIIDX

Returns the index of the entry for the specified server in the server information table.

---

## Syntax

**%SERVIIDX**(*server-name*);

The SERVIDX macro requires a server name and returns the index for that server entry in the server information table. This index can be used to access the fields in the table entry.

*Note:* *server-name* cannot be specified as an alias. △

---

## SERVINFO

Adds server attributes to the server information table.

---

### Syntax

```
%SERVINFO(two-level-server-name,
  <RMTVIEW=REMOTE-engine-RMTVIEW=-option>,
  <NETNODE=network-node-name>);
```

Typically, this macro is used in the member SERVERID of the APPLSYS macro library.

A server information table is created to contain information about the servers at your site. You can use this information in a program or you can display it. By default, the table contains the following type of information:

- a default value for the REMOTE engine's RMTVIEW= option in the LIBNAME statement
- a network node name that is represented by a fully-qualified node name; for example, HP103.DOM2.ACME.COM.

This table can also be used to specify other characteristics of a server or its users or administrators, such as server access passwords, PROC SERVER statement options, and the release of SAS software that the server is running under.

Optional arguments to the SERVINFO macro follow:

RMTVIEW=

specifies a default value for the REMOTE engine's RMTVIEW= option in the LIBNAME statement. If you specify this parameter, the LIBDEF macro generates by default the RMTVIEW=*value* for any LIBNAME statement that specifies this server. This parameter is overridden by the RMTVIEW= parameter in the SERVLIB and LIBDEF macros.

NETNODE=

specifies a network node name that is represented by the high-level qualifier in a two-level server name. When a two-level server name is specified in a PROC OPERATE or a LIBNAME statement and the high-level qualifier cannot be found as a network node, the server name is treated as the name of a macro variable whose value is the node name. This substitution is useful when the node name is not a valid SAS name. If you specify NETNODE= in the server information table, the SERVERID macro generates the following code in an application, the first time that it translates an alias for that serverid:

```
%LET high-level-qualifier=network-node;
```

where *high-level-qualifier* is the high-level qualifier in the serverid that is specified in the positional parameter and *network-node* is the value of NETNODE=.

If the high-level qualifier in the serverid is also the high-level qualifier in the full network node name, you may omit it from the value of NETNODE= by beginning the NETNODE= value with a period (.).

For example, if the server SHRSERV runs on HP1.03.DOM2.ACME.COM, you would specify

```
%servinfo (hp.shrserv,netnode=hp103.dom2.acme.com);
```

The SERVERID macro generates

```
%let hp=hp103.dom2.acme.com;
```

---

## SERVLIB

Adds server-library pairs to the library table.

---

### Syntax

```
%SERVLIB(SAS-library-name, server-name
  <RMTVIEW=REMOTE-engine-RMTVIEW=-option>,
  <PHYSNAME=physical-name-of-library>,
  <SLIBREF=server-libref> ,
  <ENGINE=engine-in-user-session> ,
  <RENGINE=engine-in-server-session>);
```

The SERVLIB macro is used to add new libraries to the library table in the APPLSYS macro library in the form *SAS-library-name*, *server-name*. How you specify the SAS library name is host-specific. Optional arguments to the SERVLIB macro follow:

#### RMTVIEW=

specifies the value of the REMOTE engine's RMTVIEW= option in the LIBNAME statement that is generated by the LIBDEF macro. The RMTVIEW= parameter in the SERVLIB macro can be used to specify the default value of this option for a specific library. This parameter overrides the RMTVIEW= parameter in the SERVINFORM macro. But, the RMTVIEW= parameter in the SERVLIB macro is overridden by the RMTVIEW=parameter in the LIBDEF macro

#### PHYSNAME=

specifies the physical name of a library. This parameter is used by the STRTSRV macro to generate a LIBNAME statement in the server session. If this parameter is used and the SLIBREF= parameter is not used, the first positional parameter is assumed to be the server libref. If both PHYSNAME= and SLIBREF= are specified, the first positional parameter is not used for generating the LIBNAME statement. Instead, the first positional parameter can be used as a description of the library that is specified in place of the physical name when the LIBDEF macro is invoked in the application.

#### SLIBREF=

specifies the library's libref in the server session. This parameter is used by the STRTSRV macro to generate a LIBNAME statement in the server session. If this parameter is used and the PHYSNAME= parameter is not used, the first positional parameter in the SERVLIB macro is assumed to be the physical name of the library. If both PHYSNAME= and SLIBREF= are specified, the first positional parameter is not used for generating the LIBNAME statement. Instead, the first positional parameter can be used as a description of the library that is specified in place of the physical name when the LIBDEF macro is invoked in the application.

#### ENGINE=

specifies the engine to be used in the user session to access the library. The default is ENGINE=REMOTE. Omit this parameter unless you need to override the engine that is chosen by SAS.

ENGINE=

specifies the engine to be used in the server session to access the library. There is no default for this option. SAS chooses an appropriate engine. Omit this parameter unless you need to override the engine that is chosen by SAS.

For more information about the SERVLIB macro, see “Associating SAS Libraries with Server Aliases (SERVLIB)” on page 82 and comments in the APPLSYS macro library.

## SETSRV

Generates a SET SERVER statement.

### Syntax

```
%SETSRV (server-name, <oapw>);
```

The SETSRV macro generates an OPERATE procedure SET SERVER statement for a SAS server that is specified by using the *server-name* argument.

*server-name*

specifies the name of a server. The server name can be an alias or an actual serverid.

*oapw*

specifies the operator-access password to be specified in the SAPW= option in the SET statement.

```
%setsrv(serv1,myoperpw);
```

## SHRMACS

Compiles other SAS/SHARE macros.

### Syntax

*Note:* The SHRMACS macro must be invoked before any other SAS/SHARE macro. △

```
%SHRMACS category, <log-info>,
           <APPLSYS=app-sys-lib-tab . . .>,
           <SASSAML=alt-sys-lib-tab...>);
```

The SHRMACS macro generates and compiles the SAS/SHARE macros according to the following parameters:

*category*

defines type of macros: USER, SERVER, OPER, or ALL. For details about the macros that are contained within these categories, see “Compiling Macros (SHRMACS)” on page 77.

*log-info*

The MSG argument displays the SAS/SHARE macros that are generated and displays what each macro does. MSG is the default action; if the first argument is

specified alone, MSG information is written to the SAS log. If the NOMSG argument is specified, no information is displayed.

The HELP argument displays detailed information about the SAS/SHARE macros that are generated. This includes the syntax, a brief description, and an example of each macro that is generated by the first argument.

#### APPLSYS=

The optional APPLSYS= argument specifies which applications systems tables are to be loaded. You can specify one or more tables. It is recommended that you use this argument to save initialization time. If the APPLSYS= argument is not specified, the default applications systems table is loaded. For more information about using the APPLSYS= argument, see “Setting Up the Application System (APPLSYS) Macro Library” on page 80.

#### SASSAML=

The optional SASSAML= argument specifies an alternate APPLSYS macro library. If an alternate library is specified, application systems tables are loaded from it instead of from the default library. The value of SASSAML= can be a physical pathname or the string `_DEFINED_`, which indicates that the fileref SASSAML is already assigned to the alternate APPLSYS macro library. For more information about the SASSAML= argument, see “Setting Up the Application System (APPLSYS) Macro Library” on page 80.

The SHRMACS macro also loads the applications systems tables that associate aliases with server names and libraries with aliases. These tables are used to generate the server name for the PROC SERVER, PROC OPERATE, and LIBNAME statements. Based on what is specified in the first argument, the server-alias and library-alias tables may be written to the SAS log if you specify the MSG or the HELP argument in the SHRMACS macro.

```
%shrmacs(user); %shrmacs(user,help);
%shrmacs(oper,help); %shrmacs(server,msg); %shrmacs(all,msg);
```

The SHRMACS macro also obtains the current settings of the NOTE, SOURCE2, and LINESIZE= SAS options in order to restore them later. SHRMACS invokes PROC SQL early in its execution to obtain the settings of these options and to save them in macro variables named `_NOTES_`, `_SRC2_`, and `_LS_`, respectively. The original values are restored after the settings have been changed by SHRMACS or other SAS/SHARE macros. You can avoid the overhead of this PROC SQL step by explicitly setting the macro variables to the desired values, as shown here:

```
%let _notes_=notes; %let _src2_=nosource2;
%let _ls_=70;
```

---

## SHUTSRV

Stops a server.

---

### Syntax

The SHUTSRV macro invokes the OPERATE procedure to terminate a server that is specified with the *server-name* argument.

```
%SHUTSRV(server-name, < oapw>);
```

*server-name*

specifies the server name. The name can be an alias or an actual serverid.



*oapw*

specifies the operator-access password to be mapped to the SAPW= option in the PROC OPERATE statement.

```
%shutsrv(serv1,myoperpw);
```

---

## STRTSRV

Starts a server.

---

### Syntax

The STRTSRV macro invokes the SERVER procedure.

```
%STRTSRV(server-name, <options>, <uapw>, <oapw>);
```

The STRTSRV macro generates a LIBNAME statement for each library in the current library table for the server being started (either directly or through an alias) for which either the SLIBREF= or PHYSNAME= parameters to SERVLIB have been specified (see “Syntax” on page 138). Using this mechanism to define a library to the server at start-up automates the change to server start-up when the library is switched to another server.

*server-name*

specifies the server name. The name can be an alias or an actual serverid.

*options*

specifies any PROC SERVER statement options that you want to use. Use blank spaces to separate options. For details about the PROC SERVER options, see Chapter 8, “The SERVER Procedure,” on page 95.

*uapw*

specifies the user-access password.

*oapw*

specifies the operator-access password to be mapped to the UAPW= and OAPW= options, respectively, in the PROC SERVER statement.

Additionally, the STRTSRV macro generates a LIBNAME statement for each library that is listed in the current library-alias table for the server being started.

```
%strtsrv(serv1,msgn noalloc,myuserpw,myoperpw);
```



The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/SHARE User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 247.

**SAS/SHARE User's Guide, Version 8**

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

ISBN 1-58025-478-0

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<sup>®</sup> 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. <sup>®</sup> indicates USA registration.

IBM<sup>®</sup>, AIX<sup>®</sup>, DB2<sup>®</sup>, OS/2<sup>®</sup>, OS/390<sup>®</sup>, RMT<sup>™</sup>, RS/6000<sup>®</sup>, System/370<sup>™</sup>, and System/390<sup>®</sup> are registered trademarks or trademarks of International Business Machines Corporation. ORACLE<sup>®</sup> is a registered trademark or trademark of Oracle Corporation. <sup>®</sup> 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.