

## CHAPTER

## 17

## The UPLOAD Procedure

---

<i>Introduction</i>	107
<i>Syntax for the UPLOAD Procedure</i>	108
<i>Syntax for the PROC UPLOAD Statement</i>	108
<i>PROC UPLOAD Statement Options</i>	109
<i>Default Naming Conventions for Uploaded Data Sets</i>	116
<i>Data Set Options and Attributes for the UPLOAD Statement</i>	118
<i>Syntax for the WHERE Statement</i>	118
<i>Syntax for the EXCLUDE Statement</i>	119
<i>Members of a SAS Data Library</i>	119
<i>Entries in a SAS Catalog</i>	120
<i>Syntax for the SELECT Statement</i>	120
<i>Members of a SAS Data Library</i>	120
<i>Entries in a SAS Catalog</i>	121
<i>Syntax for the TRANTAB Statement</i>	121
<i>Requirements for the TRANTAB Statement</i>	122
<i>Options for the TRANTAB Statement</i>	122
<i>Using the MEMTYPE= Option</i>	123
<i>Using the ENTRYTYPE= Option</i>	123
<i>Compiling PROGRAM Entries</i>	124
<i>Transfer Status Window</i>	124
<i>The BINARY Option</i>	125
<i>Example of the BINARY Option</i>	126
<i>PROC UPLOAD Output</i>	126
<i>Defining Librefs and Filerefs</i>	127
<i>Non-English Keyboards</i>	127
<i>General Tips for UPLOAD</i>	128

---

## Introduction

After you have started SAS/CONNECT software, you can transfer files between your local session and the remote host. The UPLOAD procedure sends files from the local host to the remote host.

Using PROC UPLOAD in SAS/CONNECT, you can

- transfer multiple SAS files in a single step by using the INLIB= and OUTLIB= options. This capability enables you to transfer an entire library or selected members of a library in a single PROC UPLOAD step.
- specify certain entries in a catalog or certain members in a library that should be uploaded by using the SELECT and EXCLUDE statements.

- specify a specific translation table that should be used when uploading a SAS catalog.
- use WHERE processing and SAS data set options when uploading individual SAS data sets.
- replicate certain data set attributes when you upload a data set.
- transfer data sets and catalog entries that have been modified on or after the specified date.
- transfer encrypted SAS data set files.

This chapter describes the syntax and specifications for the UPLOAD procedure. For examples that use this syntax, see Chapter 16, “Using Data Transfer Services,” on page 103, Chapter 7, “Examples of Combining Compute Services and Data Transfer Services,” on page 55, and Chapter 12, “Example of Combining Remote Library Services (RLS) and Data Transfer Services,” on page 89.

---

## Syntax for the UPLOAD Procedure

The UPLOAD procedure is controlled by the following statements:

```
PROC UPLOAD< data-set-option(s)>
  < catalog-option(s)>
  < library-option(s)>
  < external-file-option(s)>
  <AFTER=date>
  <CONNECTSTATUS=NO>;

WHERE where-expression-1
  < logical-operator where-expression-n>;

EXCLUDE list</MEMTYPE=mtype |
  ENTRYTYPE=etype>;

SELECT </MEMTYPE=mtype | ENTRYTYPE=etype>;

TRANTAB NAME=translation-table-name
  <TYPE=(etype-list)>
  <OPT=DISP SRC (DISP SRC)>;
```

The following sections give the syntax for these statements.

---

## Syntax for the PROC UPLOAD Statement

```
PROC UPLOAD< data-set-option(s)>
  < catalog-option(s)>
  < library-option(s)>
  < external-file-option(s)>
  <AFTER=date>
  <CONNECTSTATUS=NO>;
```

- *data-set-options* can be one or more of the following:
  - CONSTRAINT=NO
  - DATA=*local-SAS-data-set*
  - EXTENDSN=NO
  - INDEX=NO
  - OUT=OUT=*remote-SAS-data-set*
  - TRANIN=
  - TRANNET=
  - TRANOUT=
  - V6TRANSPORT
- *catalog-options* can be one or more of the following:
  - ENTRYTYPE=*etype*
  - EXTENDSN=NO
  - INCAT=*local-SAS-catalog*
  - OUTCAT=*remote-SAS-catalog*
- *library-options* can be one or more of the following:
  - CONSTRAINT=NO
  - EXTENDSN=NO
  - GEN=YES
  - INDEX=NO
  - INLIB=*local-SAS-library*
  - MEMTYPE=(*mtype-list*)
  - OUTLIB=*remote-SAS-library*
  - V6TRANSPORT
- *external-file-options* are the following:
  - BINARY
  - INFILE=*local-file-identifier*
  - OUTFILE=*remote-file-identifier*

---

## PROC UPLOAD Statement Options

The following options can be used with the PROC UPLOAD statement. They are shown in alphabetical order.

AFTER=*date*

specifies a modification date in the form of a numeric date value or a SAS date constant.

This option is valid for transferring data sets, catalogs, and libraries. Its use results in data sets or catalog entries being transferred only if they have been modified on or after the specified date.

*Note:* This option is available in Releases 6.09E, 6.11 TS040, and later.  $\Delta$

For example, the following statement causes the transfer of any data sets or catalog entries in the library ACCTS only if they have been modified on or after December 30, 1995.

```
proc upload inlib=accts outlib=accts
  after='30dec95'd status=no;
```

If your local session is using an earlier release of SAS that does not support this option, PROC UPLOAD produces the following message:

```
Warning: AFTER= option not supported by earlier
  release; option will be ignored.
```

*Note:* However, if the local host is running Release 6.11 TS020 or Release 6.08 TS415 through Release 6.08 TS430, the option is ignored, but no warning is displayed.  $\Delta$

## BINARY

specifies that you want to upload a binary image (an exact copy) of an external file. Use this option only for uploading external files.

The BINARY option prevents record delimiters from being inserted at each remote host record. In addition, if the remote host uses a different method of data representation, the BINARY option prevents any data translation such as conversion from ASCII to EBCDIC. See “The BINARY Option” on page 125 for more information.

## CONNECTSTATUS=NO

## CSTATUS=NO

## STATUS=NO

specifies that the status window should not be displayed while the data set, the catalog, the library, or the external file is being uploaded. By default, the UPLOAD procedure displays the Transfer Status window.

## CONSTRAINT=YES|NO

specifies if integrity constraints should be re-created on the remote host, when a SAS data set that has integrity constraints defined is uploaded. You can specify this option with the DATA= option (if you omit the OUT= option) or with the INLIB= and OUTLIB= options.

*Note:* By default, integrity constraints are re-created when you upload a SAS library or when you upload a single SAS data set and omit the OUT= option. If you specify the OUT= option with the DATA= option, the integrity constraints are not re-created.  $\Delta$

## DATA=*local-SAS-data-set*

names a SAS data set that you want to upload to the remote host from your local host. If the data set is a permanent SAS data set, you must define a libref before the PROC UPLOAD statement and specify the two-level name of the data set.

If you specify the name of a data view in the DATA= option, the materialized data is uploaded to the remote host, not the view definition.

If you do not use the DATA=, INCAT=, or INFILE= option, the last SAS data set that was created on the local host during your SAS session is uploaded.

If you use the DATA= option, you must either use the OUT= option or omit all other options.

Using the DATA= option, SAS/CONNECT supports the transfer of long-named members for data set transfers, as long as the host supports long member names.

ENTRYTYPE=*etype*

ETYPE=*etype*

ET=*etype*

specifies a catalog entry type to be uploaded. You can use this option only when you specify both the INCAT= and OUTCAT= options. See “Using the ENTRYTYPE= Option” on page 123 for more information.

EXTENDSN=YES|NO

specifies whether or not to promote the length of short numerics (length less than 8 bytes) when transferring.

NO

indicates that the length of numeric variables is not promoted. This is the default.

YES

indicates that 1 will be added to the length of any numeric variable that has a length of less than 8 bytes before it is transferred to the target host. The behavior of this option differs based on the release that is being used.

- When both the local and remote hosts are Version 7 or Version 8 and the V6TRANSPORT option is specified, the default behavior is to promote the length of numerics whose lengths are less than 8 bytes. This is consistent with Version 6 behavior. To override this behavior, specify EXTENDSN=NO along with the V6TRANSPORT option in the UPLOAD statement.
- When the local host is Version 7 or Version 8 and the remote host is Version 6, neither the V6TRANSPORT nor the EXTENDSN= option is supported or recognized.
- When the local host is Version 6 and the remote host is Version 7 or Version 8, neither the V6TRANSPORT nor the EXTENDSN= option is supported or recognized.

In Version 6 style translation, translation occurs twice for every transmission. The data is translated from local to transport format, and then the receiving host translates from transport format to local format. Two translations occur for all data that is transferred.

When both the local host and the remote host are Version 7 or Version 8, 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 running Version 7 or Version 8, the data is translated only one time when translation is necessary and is not translated when both hosts have the same machine architecture.

GEN=YES|NO

specifies that data set generations are to be sent during library transfers.

YES

data set generations are sent during library transfers. This is the default.

NO

data set generations are *not* sent during library transfers.

INCAT=*local-SAS-catalog*

names a SAS catalog that you want to upload to the remote host from your local host. If the catalog is stored in a permanent SAS data library, you must define a libref before specifying the PROC UPLOAD statement, and you must specify the catalog's two-level name. To upload all of the catalogs in a SAS data library, specify

INCAT=*libref.\_ALL\_*

If you specify this form for the INCAT= option, you must specify the same form for the OUTCAT= option.

You can transfer catalogs with entries that contain graphics output as well as other catalog entries.

Using the INCAT= option, SAS/CONNECT supports the transfer of long-named members for catalog transfers, as long as the host supports long member names.

INLIB=*local-SAS-library*

IN=*local-SAS-library*

INDD=*local-SAS-library*

names a SAS data library that you want to upload from the local host to your remote host. All three forms of this option are equivalent. This option must be used with the OUTLIB= option (in any of its forms). Before using this option, you must define the libref that is used for *local-SAS-library*.

The following examples illustrate some valid pairs of these options:

INLIB= OUTLIB=

INLIB= OUT=

IN= OUT=

IN= OUTDD=

INDD= OUTDD=

INDD= OUTLIB=

INDEX=NO

specifies that when you upload a SAS data set that has an index, the index is not re-created on the remote host. You can specify this option with the DATA= option (if you omit the OUT= option) or by using the INLIB= and OUTLIB= options. Note that the index is re-created by default when you upload a single data set and omit the OUT= option, or when you upload a SAS data library.

If you specify the OUT= option with the DATA= option, the index is not re-created. To create an index for the output data set, use the INDEX= data set option in the data set name that you specify in the OUT= option to define the index. The INDEX= data set option is described in *SAS Language Reference: Dictionary*.

INFILE=*local-file-identifier*

specifies the external file that you want to upload to the remote host from the local host; *local-file-identifier* can be one of the following:

*fileref*

is used if you have defined a fileref on the local host that is associated with a single file. You must define the fileref before specifying the PROC UPLOAD statement.

*fileref(member)*

is used if you have defined a fileref on the local host that is associated with an aggregate storage location, such as a directory. *member* specifies the

particular file(s) to transfer in that aggregate storage location. An asterisk (\*) can be used as a wildcard character in the *member* specification to transfer

- all files in the specified location (\*).
- all files that have the same extension (\*.extension).
- all files that have the same name but different extensions (name.\*).

You must define the fileref before specifying the PROC UPLOAD statement. Refer to the SAS documentation for your local host for more information about this use of filerefs.

The following example demonstrates how to use a wildcard to transfer (upload) all files that have the extension *.sas* and are located in a directory on a remote UNIX host to a folder on a local PC.

```
filename locref 'c:\';
rsubmit;
    filename fref '//local/programs';
    proc upload infile=locref('*.*.sas')
                outfile=fref;
    run;
endrsubmit;
```

*'external-file-name'*

is used to explicitly define the file that is to be uploaded.

If you use the INFILE= option, you must also use the OUTFILE= option.

MEMTYPE=(*mtype-list*)

MTYPE=(*mtype-list*)

MT=(*mtype-list*)

specifies one or more member types to be uploaded. You can specify this option only with the INLIB= and OUTLIB= options. See “Using the MEMTYPE= Option” on page 123 for more information.

OUT=*remote-SAS-data-set*

names the SAS data set on the remote host to which you want the uploaded data set written. If you want to create a permanent SAS data set, you must define the libref before specifying the PROC UPLOAD statement, and you must specify a two-level SAS data set name.

Using the OUT= option, SAS/CONNECT supports the transfer of long-named members for data set transfers, as long as the host supports long member names.

The OUT= option is a valid form of the OUTLIB= option. The UPLOAD procedure determines how to interpret the meaning of the OUT= option as follows:

- When you specify the DATA= option and the OUT= option, the OUT= option names the output SAS data set.

For example, if the USER= option is set to MYLIB, the following statement uploads the data set A from the library MYLIB on the local host to the library MYLIB on the remote host:

```
proc upload data=a out=a;
run;
```

- If you specify only the OUT= option, the UPLOAD procedure uploads the last SAS data set that was created on the local host.

For example, the following statement uploads the last data set that was created on the local host to the data set MYDATA in the library MYLIB on the remote host (assuming USER=MYLIB).

```
proc upload out=mydata;
run;
```

- When you specify the INLIB= option (or some other form of this option) with the OUT= option, the OUT= option specifies the name of a SAS data library.

For example, the following statement uploads all of the data sets and catalogs that are in the library A on the local host to the library RMTLIB on the remote host.

```
proc upload inlib=a out=rmtlib;
run;
```

See “Default Naming Conventions for Uploaded Data Sets” on page 116 for information about the effect of omitting the OUT= option.

#### OUTCAT=*remote-SAS-catalog*

names the SAS catalog on the remote host to which you want the uploaded catalog written. If you want to create a permanent SAS catalog, you must define the libref before specifying the PROC UPLOAD statement and specify a two-level SAS catalog name. To upload all of the catalogs in a SAS data library, specify

```
OUTCAT=libref._ALL_
```

If you specify this form for the OUTCAT= option, you must specify the same form for the INCAT= option.

If you use the OUTCAT= option, you must also use the INCAT= option.

Using the OUTCAT= option, SAS/CONNECT supports the transfer of long-named members for catalog transfers, as long as the host supports long member names.

#### OUTFILE=*remote-file-identifier*

identifies an external file on the remote host to which you want an uploaded external file written. *remote-file-identifier* can be one of the following:

##### *fileref*

is used if you have defined a fileref on the remote host that is associated with a single file. You must define the fileref before specifying the PROC UPLOAD statement.

##### *fileref(member)*

is used if you have defined a fileref on the remote host that is associated with an aggregate storage location, such as a directory or a partitioned data set. *member* specifies the particular file in that aggregate storage location. You must define the fileref before specifying the PROC UPLOAD statement. Refer to the SAS documentation for your remote host for more information about this use of filerefs.

*Note:* If a wildcard (\*) is used in the INFILE= option, then OUTFILE=*fileref* should point to an aggregate storage location such as a directory.  $\Delta$

##### *'external-file-name'*

is used to explicitly define the file that is to be uploaded.

If you use the OUTFILE= option, you must also use the INFILE= option.

#### OUTLIB=*remote-SAS-library*

#### OUT=*remote-SAS-library*

#### OUTDD=*remote-SAS-library*

names the destination SAS data library on your remote host where the uploaded data sets and catalogs from the local host are stored. All three forms of this option



are equivalent. Before using this option, you must define the libref that is used for *remote-SAS-library*.

*Note:* The OUT= form of this option is the same as the OUT= option that is used to specify a SAS data set. When you use this option, the UPLOAD procedure determines whether the input option was DATA= or INLIB= and processes the uploaded objects appropriately. △

The OUTLIB= option must be used with the INLIB= option, but you can use any form of the OUTLIB= option with any form of the INLIB= option. Refer to the description of the INLIB= option for examples that illustrate some valid pairs of these options.

**TRANIN=***translation-table-name*

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

For example, you used FTP to transfer a data set from a UNIX machine to your local PC, and now you want to transfer the data set for use on a VAX. The data set is not in the native PC format (your local host), so the default translation table will not work. Using the TRANIN= option with PROC UPLOAD, you can specify a translation table that transfers the data from UNIX to the VAX.

The translation will take place on the local host before the data is transferred to the remote host. This option is equivalent to using the TRANTAB= data set option on the input data set.

*Note:* This option is only valid for data set transfers between two Version 7 or Version 8 sessions and when the transfer is between two hosts that have different machine architectures, which makes translation necessary. △

**TRANNET=***translation-table-name*

specifies a translation table to be used when transferring a data set between local and remote hosts with different machine architectures. The translation table specified will be used in place of the default translation table. The translation will take place on the remote host after the data is transferred from the local host.

*Note:* This option is only valid for data set transfers between two Version 7 or Version 8 sessions and when the transfer is between two hosts that have different machine architectures, which makes translation necessary. △

**TRANOUT=***translation-table-name*

specifies a translation table to be used when transferring a local data set to a remote host, and the OUTREP= data set option indicates a data representation different from the native data representation of the remote host. The translation will take place on the remote host after the data is transferred from the local host.

For example, you are using a PC and would like to transfer a data set to a UNIX machine, and the OUTREP=VAX\_VMS data set option is specified in a DATA statement. Using the TRANOUT= option with PROC UPLOAD, you can specify a translation table that transfers the data set from your local PC to the UNIX machine in a VAX format.

*Note:* This option is only valid for data set transfers between two Version 7 or Version 8 sessions and when the transfer is between two hosts that have different machine architectures, which makes translation necessary. △

**V6TRANSPORT**

specifies that data should be translated in the Version 6 style. This option should only be specified when the Version 6 translation style is desired and both the local and remote hosts are Version 7 or Version 8. Version 6 style translation means

that the data is translated from a local format to a transport format, and then the receiving host translates the data from the transport format to a local format.

When V6TRANSPORT is specified, the default behavior is to promote the length of numerics that are less than 8 bytes. The EXTENDSN=NO option may be used with the V6TRANSPORT option to prevent the lengths from being promoted.

When transferring files from Version 7 or Version 8 to Version 6, SAS/CONNECT checks that the files are Version 6 compliant. If the files are not Version 6 compliant, an error message is issued and the transfer is terminated. For example, a data set from a Version 7 or Version 8 session that contains a variable with a long name will not transfer because Version 6 does not support long variable names.

Data set and variable names are truncated if they exceed the Version 6 maximum. A warning message is issued about the truncation, but the transfer will continue.

In Version 6, the value of a character variable has a maximum length of 200. For Version 7 or Version 8, this maximum has been increased to 32K. Therefore, a Version 7 or Version 8 variable value that has a length greater than 200 will transfer to a Version 6 variable, but the value will be truncated to 200. A warning message is issued about the truncation, but the transfer will continue.

---

## Default Naming Conventions for Uploaded Data Sets

Follow these rules when you upload a SAS data set and do not specify the name of the data set in the OUT= option:

- If the input data set (the data set that is specified in the DATA= option) has a two-level name and the same libref that is defined for the input data set is also defined in the remote host environment, the data set is uploaded to the library on the remote host that is associated with that libref. The data set has the same member name on the remote host.

For example, suppose you submit the following statement:

```
libname orders
    local-host-SAS-data-library;
```

When you remote submit the following statements, the data set ORDERS.QTR1 is uploaded to ORDERS.QTR1 on the remote host.

```
/* ***** */
/* The libref ORDERS is defined on both */
/* hosts. */
/* ***** */
libname orders
    remote-host-SAS-data-library;
proc upload data=orders.qtr1;
run;
```

- If the input data set has a two-level name but the libref for the input data set is not also defined in the remote host environment, the data set is uploaded to the default library on the remote host. This is usually the WORK library, but the library may also be defined by using the USER libref.

The data set retains the same data set name that it had on the local host. For example, if you remote submit the following statement, the data set is uploaded to WORK.QTR2 on the remote host.

```
/* ***** */
/* The libref ORDERS is defined only on */
/* the local host. */
```

```

          /*****/
proc upload data=orders.qtr2;
run;

```

- If the input data set has a one-level name and the default libref on the local host also exists on the remote host, the data set is uploaded to that library.

For example, suppose you submit the following statements:

```

libname orders
  local-host-SAS-data-library;
options user=orders;

```

If you remote submit the following statements, the data set ORDERS.QTR1 is uploaded to ORDERS.QTR1 on the remote host.

```

          /*****/
/* The libref ORDERS is defined on both */
/* hosts.                               */
          /*****/
libname orders
  remote-host-SAS-data-library;
libname remote
  remote-host-SAS-data-library;
          /*****/
/* This option has no effect in        */
/* this case.                          */
          /*****/
options user=remote;
proc upload data=qtr1;
run;

```

- If the input data set has a one-level name and the default libref on the local host does not exist on the remote host, the data set is uploaded to the default library on the remote host. That is, the USER libref on the remote host is used only if the USER libref on the local host does not exist on the remote host.

For example, suppose you submit these statements:

```

libname orders
  local-host-SAS-data-library;
options user=orders;

```

When you remote submit the following statements, the data set ORDERS.QTR1 is uploaded to REMOTE.QTR1 on the remote host.

```

          /*****/
/* The libref ORDERS is defined only on */
/* the remote host.                     */
          /*****/
libname remote
  remote-host-SAS-data-library;
options user=remote;
proc upload data=qtr1;
run;

```

- If you omit the DATA= option, as in this example,

```

proc upload;
run;

```

the last data set that was created on the local host during the SAS session is uploaded to the remote host.

The naming conventions on the remote host follow one of the rules described above, based on how the last data set that was created was named.

---

## Data Set Options and Attributes for the UPLOAD Statement

PROC UPLOAD permits you to specify SAS data set options in the DATA= and OUT= options. Note that SAS data set options are not supported with the INLIB= and OUTLIB= options, even when you upload only data sets. The data set options must be associated with a specific SAS data set, so they must be used in the DATA= or OUT= options.

In addition, when you upload SAS data sets by using the DATA= option (omitting the OUT= option) or the INLIB= and OUTLIB= options, or if you omit all of these options, the following characteristics are inherited by the uploaded data set.

*Note:* The following list of characteristics shows the SAS data set option (in parenthesis) that was used to create the characteristic for the original data set. You do not have to specify the option to have it inherited when the data set is uploaded.  $\Delta$

- password for ALTER protection (ALTER= SAS data set option)
- compressed observations (COMPRESS= SAS data set option)
- indexes (INDEX= SAS system option or other methods of creating indexes).

*Note:* The index for an uploaded SAS data set is re-created on the remote host; not copied from the local host. To prevent the re-creation of the index, you can specify the INDEX=NO option in the PROC UPLOAD statement, as described in “PROC UPLOAD Statement Options” on page 109.  $\Delta$

- data set label (LABEL= SAS data set option)
- password for READ protection (READ= SAS data set option)
- re-use of free space in compressed data sets (REUSE= SAS data set option)
- list of variables that the data set is sorted by (SORTEDBY= SAS data set option)
- data set type (TYPE= SAS data set option)
- password for WRITE protection (WRITE= SAS data set option)
- integrity constraints
- maximum generations.

If you specify the OUT= option when uploading a single data set, only the following characteristics are inherited by the uploaded data set:

- data set type
- data set label.

The following example illustrates the use of the DATA= option and the INDEX=NO option. The example also shows the use of the KEEP= SAS data set option. Note that because no OUT= option is specified, the uploaded data set inherits the characteristics of the input data set except the index (because the INDEX=NO option is specified).

```
proc upload data=study(keep=age score1 score2) index=no;
run;
```

---

## Syntax for the WHERE Statement

**WHERE** *where-expression-1*  
 < *logical-operator where-expression-n*>;

The UPLOAD procedure processes WHERE statements and the WHERE= data set option when you transfer a single SAS data set. WHERE statements allow multiple WHERE expressions that are joined by logical operators. *logical-operator* can be OR, OR NOT, AND, or AND NOT.

You can use SAS functions in a WHERE expression. Also, note that a DATA or a PROC step attempts to use an available index to optimize the selection of data when an indexed variable is used in combination with one of the following:

- CONTAINS operator
- LIKE operator
- colon modifier with a comparison operator
- TRIM function
- SUBSTR function (in some cases).

To understand when use of the SUBSTR function causes an index to be used, look at the format of the SUBSTR function in a WHERE statement:

```
where substr(variable, position, length)
      ='character-string';
```

An index is used in processing when all of the following conditions are met:

- *position* is equal to 1
- *length* is less than or equal to the length of *variable*
- *length* is equal to the length of *character-string*.

The following example illustrates using a WHERE statement with the UPLOAD procedure. The uploaded data set contains only the observations that meet the WHERE condition.

```
proc upload data=revenue out=new;
  where origin='Atlanta' and revenue < 10,000;
run;
```

For more information about the WHERE statement, refer to *SAS Language Reference: Dictionary*.

## Syntax for the EXCLUDE Statement

**EXCLUDE** *list*</ MEMTYPE=*mtype* |  
 ENTRYTYPE=*etype*>;

The EXCLUDE statement enables you to exclude from uploading specific members in a SAS data library or specific entries in a SAS catalog. You cannot use both the EXCLUDE and SELECT statements in the same PROC UPLOAD step, but you can specify multiple EXCLUDE statements.

The EXCLUDE statement can have two forms, which are discussed in detail in the next two sections.

### Members of a SAS Data Library

You can use the following form of the EXCLUDE statement with the INLIB= and OUTLIB= options in the PROC UPLOAD statement:

**EXCLUDE** *list*</ MEMTYPE=*mtype*>;

When you use the EXCLUDE statement to exclude members of a library from uploading, *list* specifies the members to exclude from uploading. You can explicitly name all members to exclude or use one of the following forms of name lists:

- : (colon)
  - specifies all member names that begin with the string of characters that immediately precede the colon. For example, if you specify **TEST:**, none of the members that begin with the letters **TEST** are uploaded.
- (hyphen)
  - specifies all member names that fall lexicographically between the names on either side of the hyphen. You can use this wild-card character only between two names that begin with the same string of characters and end in a number, for example, **TEST1-TEST5**.

For information about using the MEMTYPE= option, see “Using the MEMTYPE= Option” on page 123.

## Entries in a SAS Catalog

You can use the following form of the EXCLUDE statement with the INCAT= and OUTCAT= options in the PROC UPLOAD statement:

**EXCLUDE** *list*</ ENTRYTYPE=*etype*>;

When you use the EXCLUDE statement to exclude catalog entries from uploading, each element of *list* has the form *entry.type*. In this form,

- entry* is the name of an entry in the *catalog* to exclude from uploading.
- .type* is the type of the catalog entry. This part of the name is optional.

The ENTRYTYPE= option in the EXCLUDE statement works like the ENTRYTYPE= option in the PROC UPLOAD statement. For a complete list of entry types and for detailed information about how to use the ENTRYTYPE= option, see “Using the ENTRYTYPE= Option” on page 123.

---

## Syntax for the SELECT Statement

**SELECT** *list*</ MEMTYPE=*mtype* |  
ENTRYTYPE=*etype*>;

The SELECT statement enables you to select for uploading specific members in a SAS data library or specific entries in a SAS catalog. You cannot use both the EXCLUDE and SELECT statements in the same PROC UPLOAD step, but you can specify multiple SELECT statements.

The SELECT statement can have two forms, which are discussed in detail in the next two sections.

### Members of a SAS Data Library

You can use the following form of the SELECT statement with the INLIB= and OUTLIB= options in the PROC UPLOAD statement:

**SELECT** *list*</ MEMTYPE=*mtype*>;

When you use the SELECT statement to select members of a library for uploading, *list* specifies the members to select for uploading. You can explicitly name all members to select or use one of the following forms of name lists:

- : (colon)
  - specifies all member names that begin with the string of characters that immediately precede the colon. For example, if you specify **TEST:**, all of the selected members begin with the letters **TEST**.
- (hyphen)
  - specifies all member names that fall lexicographically between the names that are on either side of the hyphen. You can use this wild-card character only between two names that begin with the same string of characters and end in a number, for example, **TEST1-TEST5**.

For information about using the MEMTYPE= option, see “Using the MEMTYPE= Option” on page 123.

## Entries in a SAS Catalog

You can use the following form of the SELECT statement with the INCAT= and OUTCAT= options in the PROC UPLOAD statement:

```
SELECT list</ ENTRYTYPE=etype>;
```

When you use the SELECT statement to select catalog entries for uploading, each element of *list* has the form *entry.type*. In this form,

- entry* is the name of an entry in *member* to select for uploading.
- .type* is the type of the catalog entry. This part of the name is optional.

The ENTRYTYPE= option in the SELECT statement works like the ENTRYTYPE= option in the PROC UPLOAD statement. For a complete list of entry types and for detailed information about how to use the ENTRYTYPE= option, see “Using the ENTRYTYPE= Option” on page 123.

*Note:* The SELECT statement also enables you to maintain an ordering and grouping of catalog entries that contain graphics output, because entries are uploaded into the remote SAS catalog in the order that you specify them in the SELECT statement.

See Chapter 16, “Using Data Transfer Services,” on page 103 for an example that illustrates how to use the SELECT statement to maintain the order and grouping of catalog entries that contain graphics output. △

---

## Syntax for the TRANTAB Statement

```
TRANTAB NAME=translation-table-name
      <TYPE=(etype-list)>
      <OPT=DISP SRC (DISP SRC)>;
```

The TRANTAB statement enables you to specify translation tables to use for translating characters in catalog entries that you want to upload. The TRANTAB statement also enables you to specify the particular types of SAS/AF or SCL catalog

entries to which you want to apply a translation. You can specify one translation table per statement, but there is no limit to the number of TRANTAB statements you can use in one invocation of the UPLOAD procedure. If you specify the TRANTAB statement, you must also specify the INCAT= and OUTCAT= options in the PROC UPLOAD statement.

---

## Requirements for the TRANTAB Statement

A TRANTAB statement must include the following argument:

NAME=*translation-table-name*

specifies the name of the translation table to apply to the SAS catalog that you want to upload. The *translation-table-name* that you specify is the name of a catalog entry in either your SASUSER.PROFILE catalog or the SASHELP.HOST catalog. The UPLOAD procedure searches the SASUSER.PROFILE catalog first, and the SASHELP.HOST catalog second.

See “TRANTAB=” on page 224 for more information about translation tables.

By default, whenever you invoke PROC UPLOAD, the procedure uses the table that translates host-to-transport format. In most cases, the default translation table is the correct one to use, but you may need to apply additional translation tables if, for example, your application requires different national language characters.

You can specify a translation table other than the default in two ways:

- To specify a translation table for an invocation of PROC UPLOAD, use the TRANTAB statement in the UPLOAD procedure.
- To specify a translation table for your entire SAS session or job (including all file transfers that you do), use the TRANTAB= system option. You use this system option to define which default translation tables to use in place of those that are supplied by SAS Institute.

See the *SAS Procedures Guide* for information about creating translation tables by using PROC TRANTAB.

---

## Options for the TRANTAB Statement

You can use the following options in the TRANTAB statement:

TYPE=(*etype-list*)

specifies that the UPLOAD procedure apply the translation table only to the entries with the type or types that you specify. Note that *etype-list* can be one or more entry types. If *etype-list* is a simple entry type, omit the parenthesis.

By default, the UPLOAD procedure applies the translation table to all of the entries in the catalog that you specify.

OPT=DISP SRC (DISP SRC)

specifies one of the following:

- When OPT=DISP, the UPLOAD procedure applies the translation table only to those entries of the catalog that you specified, which produce window displays.
- When OPT=SRC, the UPLOAD procedure applies the translation table only to those entries of the catalog you specified that are of the type SOURCE.
- When OPT=(DISP SRC), the UPLOAD procedure applies the translation table only to those entries of the catalog that you specified that either produce window displays or have a type of SOURCE.

When you do not specify the OPT= option, the UPLOAD procedure applies the translation table to all of the entries in the catalog that you specify.



---

## Using the MEMTYPE= Option

If you specify the INLIB= and OUTLIB= options in the PROC UPLOAD statement, you can use the MEMTYPE= option to indicate what member types to upload. You can specify the MEMTYPE= option in

- the PROC UPLOAD statement.
- the SELECT or the EXCLUDE statement in parenthesis immediately after a member name. If the type is enclosed in parenthesis, it refers only to the member name that immediately precedes the option.
- the SELECT or the EXCLUDE statement after a slash (/) at the end of the statement. When used following a slash, the MEMTYPE= option refers to all members that are named in the statement unless the same option appears in parenthesis after a name.

Valid values for the MEMTYPE= option are DATA, CATALOG (or CAT), MDDB, FDB, DMDB, VIEW, and ALL.

If you use this option in the SELECT and EXCLUDE statements, you can specify only one value. If you use this option in the PROC UPLOAD statement, you can specify a list of values enclosed in parenthesis, as shown here:

```
proc upload inlib=this outlib=that
  memtype=(data catalog);
```

You can use the MEMTYPE= option in multiple places. When you do this, the UPLOAD procedure determines the type of each member

- 1 by the value of the option in parenthesis immediately following the member name, if present.
- 2 otherwise, by the value of the option after the slash in the SELECT or the EXCLUDE statement, if present.
- 3 otherwise, by the value of the option in the PROC UPLOAD statement, if present.
- 4 If you do not specify a member type, the UPLOAD procedure uses the default type, ALL.

See “Purpose” on page 155 for examples that illustrate the use of the MEMTYPE= option.

---

## Using the ENTRYTYPE= Option

When you specify the INCAT= and OUTCAT= options in the PROC UPLOAD statement, use the ENTRYTYPE= option in the PROC UPLOAD statement, the SELECT statement, or the EXCLUDE statement to specify which entry types to upload. The ENTRYTYPE= option is required only if you use a SELECT or an EXCLUDE statement. If you omit the entry type and also omit the SELECT and EXCLUDE statements, all catalog entries are uploaded. You can specify a single entry type in

- the PROC UPLOAD statement, specify the ENTRYTYPE= option.
- the SELECT or the EXCLUDE statement, specify the ENTRYTYPE= option in parenthesis immediately after an entry name. If the type is enclosed in parenthesis, it refers only to the entry name that immediately precedes the option.
- the SELECT or the EXCLUDE statement, specify the ENTRYTYPE= option after a slash (/) at the end of the statement. When used following a slash, the ENTRYTYPE= option refers to all entries that are named in the statement unless the same option appears in parenthesis after an entry name.

- the SELECT or the EXCLUDE statement, specify the entry type as part of the entry name. For example, you can specify TEST.MENU in the list of entries.

You can use the ENTRYTYPE= option in multiple places. When you do this, the UPLOAD procedure determines the type of each entry as described below:

- 1 by the entry type given as part of the entry name, such as XYZ.FORMAT.
- 2 by the value of the option in parenthesis immediately following the entry name, if present.
- 3 otherwise, by the value of the option after the slash in the SELECT or the EXCLUDE statement, if present.
- 4 otherwise, by the value of the option in the PROC UPLOAD statement, if present.

*Note:* If you use the ENTRYTYPE= option in both the PROC UPLOAD statement and a SELECT or an EXCLUDE statement, the values must be the same. In most cases, if you are using the ENTRYTYPE= option in a SELECT or an EXCLUDE statement, you should omit it from the PROC UPLOAD statement.  $\Delta$

For releases after Release 6.08, the following message is displayed if you try to upload an entry that is not compatible with an earlier release:

```
ERROR:  Catalog upload to earlier version not
        allowed. Backward compatibility not
        supported.
```

See “Example 2. DTS: Transferring Specific Member Types by Using SELECT or EXCLUDE Statements” on page 155 for examples of using the ENTRYTYPE= option.

## Compiling PROGRAM Entries

If you upload a catalog that contains entries of type PROGRAM, you must compile the entries on the remote host before execution. To compile all the PROGRAM entries in a catalog, remote submit the following statements to the SAS System:

```
proc build cat=libref.member-name batch;
    compile;
run;
```

where *libref* identifies the SAS data library that contains the catalog, and *member-name* identifies the catalog.

## Transfer Status Window

The Transfer Status window displays information that describes the status of the upload process. The display on the window changes as the upload proceeds. The information about the display includes:

- The type of file that is being uploaded (SAS data set, SAS catalog, catalog entry that contains graphics output, external file, or SAS utility file).
- The name of the SAS data set, SAS catalog, external file, or SAS utility file that is being uploaded. SAS data set names are displayed in the form *libref.SAS-data-set*. SAS catalogs are displayed in the form *libref.SAS-catalog*. External file names are displayed with the complete filename. Utility file names are displayed in the form *libref.SAS-utilityfilename*.
- The number of the byte that is being uploaded (updated as each new buffer is sent).

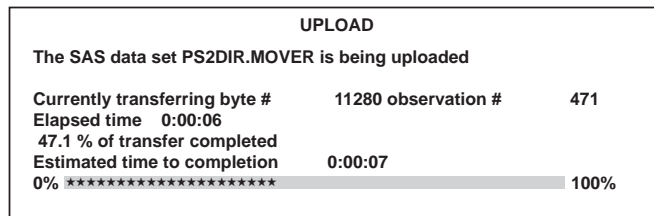
- The number of the observation that is being uploaded (for SAS data sets only).
- The time that elapsed since the beginning of the transfer, in *hh:mm:ss* form.
- The percentage of the file that is already uploaded.
- An estimate of the amount of time that is required to complete the upload, in *hh:mm:ss* form.
- A horizontal bar chart that depicts the percentage of the file that is already uploaded.

*Note:* For some types of files, the percentage completed, estimated time to completion, and the bar chart are not always available. Some operating environments cannot efficiently provide the size of the file, which is necessary to calculate these estimates.

In other cases, the information that is provided by the operating environment results in estimates that are greater than the actual time that is needed for the transfer. Thus, the percentage completed, the estimated time to completion, and the bar chart may show exaggerated estimates, but they will show 100% when the transfer is completed.  $\Delta$

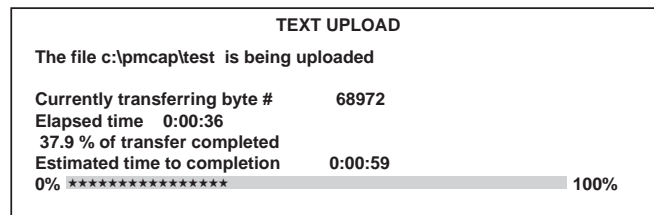
The following display is a sample of the Transfer Status window during a SAS data set upload. The SAS data set name is PS2DIR.MOVER.

**Display 17.1** Transfer Status Window for Uploading a SAS Data Set



The following display is a sample of the Transfer Status window during an external text file upload. The OS/2 pathname is C:\PMCAP\TEST.

**Display 17.2** Transfer Status Window for Uploading an External File




---

## The BINARY Option

Usually, SAS/CONNECT assumes that external files that are being uploaded are text files, and it alters them so that they can be used on the target machine. Two features are affected.

- Each operating environment has its own type of record delimiters in external files. The record delimiters are incompatible between operating environments.

Therefore, record delimiters appropriate to the target environment are added to the copied file by the UPLOAD procedure.

- CMS and OS/390 hosts use the EBCDIC character set; other operating environments use an ASCII character set. Consequently, if files are uploaded from an ASCII-based environment to CMS or OS/390 without character translations, the uploaded files cannot be interpreted by the target environment.

SAS/CONNECT solves the EBCDIC/ASCII problem by converting files automatically. The UPLOAD procedure converts text to the format of the remote host.

There are times when these file conversions are not desirable. For example, you may need to upload executable files from the local host to the remote host and later download them to the same or a different local host. You may want to do this for backup purposes or to send files to other users.

When you do not want a file translated to another character set or record delimiters inserted, you can use the BINARY option to prevent automatic conversion. When you specify the BINARY option in a PROC UPLOAD statement, SAS/CONNECT transfers the file in *binary* image form (hence, the keyword BINARY).

---

## Example of the BINARY Option

This example uses a PROC UPLOAD statement for uploading an external file to the remote host. Note that the BINARY option is included to suppress character-set translation and to prevent record-delimiter insertion:

```
proc upload infile='external-file-name'
           outfile=hostmod binary;
run;
```

This PROC UPLOAD step is excerpted from “Example 10. DTS: Distributing an .EXE File from the Remote Host to Multiple Local Hosts” on page 165.

The next example is a PROC UPLOAD statement for uploading a local SAS data set to the remote host:

```
proc upload data=pcnote.usage out=hostnote.usage;
run;
```

This PROC UPLOAD step is excerpted from “Example 12. DTS: Downloading a Partitioned Data Set from an OS/390 Host” on page 167.

---

## PROC UPLOAD Output

The UPLOAD procedure writes a series of informative messages to the SAS log when it executes. Examples of these messages are shown in the following output.

**Output 17.1** SAS Log Messages from the UPLOAD Procedure

```

NOTE: Remote submit to B commencing.
1   proc upload infile='local-external-file'
2       outfile='remote-external-file';run;

NOTE: TEXT upload in progress from infile=local-external-file
      to outfile=remote-external-file
NOTE: Uploaded 4 records and 136 bytes.
NOTE: 4 records were read from the file local-external-file
      The maximum record length was 65.
      The minimum record length was 0.
NOTE: 136 bytes were transferred at 68 bytes/second.
NOTE: The PROCEDURE UPLOAD used 0.04 CPU seconds and 1431K.

NOTE: Remote submit to B complete.
$

```

---

## Defining Librefs and Filerefs

As is true in all SAS programming, permanent SAS data sets, SAS catalogs, and external files must be defined correctly in order to be accessed. Defining a file means that you associate a shorthand, symbolic name with a file. The method that you use to define a file is based on the operating environment and whether it is a SAS file or an external file.

- Librefs for permanent SAS data libraries are defined by using the LIBNAME statement.
- Filerefs for external files are defined by using the FILENAME statement.

For more information about the LIBNAME and FILENAME statements, refer to *SAS Language Reference: Dictionary*.

Here are some special considerations to be aware of when you define files to use with the DOWNLOAD and UPLOAD procedures:

- Statements that define librefs and filerefs for files on the local host must be executed by using SUBMIT in the local SAS session. They cannot be executed by using an RSUBMIT command or an RSUBMIT statement along with the PROC DOWNLOAD or PROC UPLOAD steps.
- Statements that define librefs or filerefs for files on the remote host must be executed on the remote host by using the RSUBMIT command or the RSUBMIT statement. Therefore, these statements can be executed along with the PROC DOWNLOAD or the PROC UPLOAD step, if they are specified before the PROC statement.

---

## Non-English Keyboards

If you use a local host that has a non-English keyboard, you probably have some external files that contain non-English characters. If your remote host is CMS or TSO, some specially accented characters may be translated incorrectly when using the

DOWNLOAD and UPLOAD procedures. This occurs because of the default translations from ASCII to EBCDIC and from EBCDIC to ASCII. To solve the problem, you can do one of the following:

- If SAS/CONNECT is used frequently, you should use an alternate EBCDIC/ASCII translation table (TRANTAB=) on the remote host. The SAS Software Representative for the remote host should create the alternate table.
- If SAS/CONNECT is not used frequently, you can manage problematic characters by assigning the correct hex values in DATA step programming statements after the file is copied.

For example, suppose you have a German keyboard and a CMS host. You want a file to contain umlaut-A characters after an upload. By default, the ASCII representation of umlaut-A, which is X'84', is translated to EBCDIC X'24'. However, the EBCDIC representation of umlaut-A is X'C0', so you need to translate EBCDIC X'24' to EBCDIC X'C0'. The following DATA step, in which NAME is a variable that contains umlaut-A characters, performs this translation:

```
data new;
  set old;
  retain to 'C0'x from '24'x;
  drop to from;
  name=translate(name,to,from);
run;
```

---

## General Tips for UPLOAD

- You cannot upload a SAS data set to an external file or an external file to a SAS data set.
- If you upload to a remote host file that is defined with a fixed (F) record format, all records in the file are padded with blanks to the logical record length.
- Remember that the UPLOAD procedure executes on the remote host SAS System. You must use the RSUBMIT command, not the SUBMIT command, to execute the PROC UPLOAD step.
- The rate at which files are transferred varies based on the size and number of files that are being transferred and the processing load on the remote host, as well as the communication access method that is being used and the network configuration. The Transfer Status window keeps you informed of the progress of the transfer. See “Transfer Status Window” on page 124 for more information.
- If PROC UPLOAD successfully completes the file transfer, the macro variable SYSINFO is set to 0. If the file transfer is not successfully completed, the macro variable SYSINFO is set to a value greater than 0. You can pass the value of the SYSINFO macro variable back to the local host by using the %SYSRPUT statement. See “%SYSRPUT Statement” on page 29 for more details.
- To upload a text file with a record length that is greater than 132 bytes, the option LRECL must be specified in both the local and remote FILENAME statements.

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.