PROC DOCUMENT
(Continued)

Entry Management Statements

copy path-1, ..., path-m to path-n;
Copies path-1, ..., path-m to path-n.

move path-1, ..., path-m to path-n;
Moves path-1, ..., path-m to path-n.

make path-1, ..., path-n;
Creates new directories named path-1, ..., path-n.

import data=data-set-name to path;
Imports data set, data-set-name, or grseg, grseg-name, to path, respectively.

note path "text";
Creates a new note at path.

link path-1 to path-2;
Creates a symbolic link from path-1 to path-2.

rename path-1 to path-2;
Renames path-1 to path-2.

setlabel path "text";
Sets the label of path.

delete path-1, ..., path-n;
Deletes path-1, ..., path-n.

obbtile<n> path "text";
obanote<n> path "text";
obtitle<n> path "text";
obfootn<n> path "text";
Sets the n-th before note, after note, title, subtitle, or footnote of path, respectively. If no string is specified, the note, title, or footnote is deleted.

ohpage </<after> <delete>>;
Controls page breaks. The after option controls page breaks after path. The delete option deletes page breaks from path.

Example

* Create a new document;
ods document name=mydocument(write);

* Run procedures;
proc contents data=sashelp.class; run;

* Close the document destination;
ods document close;

* Modify and replay the document;
proc document name=mydocument;

* List the document to see
  * what paths exist;
  list / levels=all;
  run;

* Move the table we want to the
  * top level, remove titles, and
  * set a new label;
  move \Contents\DataSet\Variables to ^;
  setlabel Variables "Variable Info";
  obtitle Variables;
  obfootn Variables;
  run;

* Remove tables and directories
  * we don't need;
  delete \Contents;
  run;

* Import the dataset as a table and
  * set a new label;
  import data=sashelp.class to Class;
  setlabel Class "Student Statistics";
  run;

* List the document to verify changes;
  list / levels=all;
  run;

* Open the PDF destination;
  ods pdf;

* Replay table labeled "% Statistics";
  replay ^
    (where=(_label_ like "% Statistics"));
  run;

* Close the PDF destination;
  ods pdf close;

quit;
Output Objects

Output objects include tables, graphs, notes, and equations. In the case of tables, the output object consists of a data and a template name.

ods trace on;
Displays information about output objects as they are created. This information includes the name of the object, the name of the template, labels, paths, and label paths.

context
The context of an output object is a set of attributes that are attached to the output object. The parts of the output object context that come before the output object are: before page break, titles, subtitles, bygroup, and before notes. The parts of the output object context that come after the output object are: afternotes, footnotes, and after page break.

Content of a Document

The content of the document includes: output object, the output object context, and the output object hierarchy.

The content of the document does not include: proc options, system options, ODS options, graph options, GRSEGs, and external graph customizations.

Documents Window

odsdocuments
Entering odsdocuments into the command bar opens the Documents window, where you can view and edit the hierarchy and replay output objects. Right-clicking and choosing Properties displays information about the output object such as: type, name, description, template, the document that the output object belongs to, path in the hierarchy, file size, creation date and time, and last modified date and time.

ODS DOCUMENT and PROC DOCUMENT Tip Sheet

Documents Window

odsdocuments
Entering odsdocuments into the command bar opens the Documents window, where you can view and edit the hierarchy and replay output objects. Right-clicking and choosing Properties displays information about the output object such as: type, name, description, template, the document that the output object belongs to, path in the hierarchy, file size, creation date and time, and last modified date and time.

Document Paths

Each output object has a path associated with it that includes the directory hierarchy, much like in a computer file system. For example:

```ods trace on;
... procedure code ...
do ods document close;
document-name is the name of a document with an optional SAS® library name.
An access option can be applied to the document, document-name, by appending one of the following, in parentheses, to document-name.
write
Creates a new document. A document is overwritten if it already exists.
update
Creates a new document. A document is appended to if it already exists.
```

WHERE Clauses

Document paths can be followed by a WHERE clause to further subset the objects matched by a path. The general form of WHERE clause usage is shown below.

```document-path (where= (where-expr))```

Variables available in a WHERE clause are listed in the following table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the item</td>
</tr>
<tr>
<td>path</td>
<td>Path of the item</td>
</tr>
<tr>
<td>label</td>
<td>Label of the item</td>
</tr>
<tr>
<td>labelpath</td>
<td>Labels of all entries in path</td>
</tr>
<tr>
<td>type</td>
<td>Type of the item</td>
</tr>
<tr>
<td>seqnum</td>
<td>Sequence number of the item</td>
</tr>
<tr>
<td>cdate</td>
<td>Creation date of the item</td>
</tr>
<tr>
<td>mdate</td>
<td>Modification date of the item</td>
</tr>
<tr>
<td>ctime</td>
<td>Creation time of the item</td>
</tr>
<tr>
<td>ntime</td>
<td>Modification time of the item</td>
</tr>
<tr>
<td>cdatetime</td>
<td>Creation date-time of the item</td>
</tr>
<tr>
<td>mdatetime</td>
<td>Modification date-time of the item</td>
</tr>
<tr>
<td>variable</td>
<td>BY variable</td>
</tr>
</tbody>
</table>

Creating a Document

```ods document name=document-name;
... procedure code ...
do ods document close;
document-name is the name of a document with an optional SAS® library name.
An access option can be applied to the document, document-name, by appending one of the following, in parentheses, to document-name.
write
Creates a new document. A document is overwritten if it already exists.
update
Creates a new document. A document is appended to if it already exists.
```

Replaying a Document

```proc document name=document-name;
    replay <path-1, ..., path-n> </levels=n>;
    run;
```

Replays all output objects in a document, document-name, to all open ODS destinations.

```replay path-1, ..., path-n;
    replays only specified paths, path-1, ..., path-n, to all open ODS destinations.
```

```replay /levels=n;
    replays all output objects in document-name, within n levels of directories, to all open ODS destinations.
```

```replay path-1, ..., path-n /levels=n;
    replays only specified paths, path-1, ..., path-n, within n levels of directories to all open ODS destinations.
```

PROC DOCUMENT

Basic Usage

```proc document name=document-name;
    run;
```

Document Management Statements

doc library=library-name;
Lists all documents in library-name.

doc name=document-name;
Opens document-name for update.

doc close;
Closes the current document.

delete document-name;
Deletes document-name.

Navigation Statements

dir;
Displays the path of the current directory.

dir path;
Changes the current directory to path.

list <option(s)>;
Lists the content of the current directory or path-1, ..., path-n, respectively. The following options are available.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bygroups</td>
<td>Displays BY variables</td>
</tr>
<tr>
<td>details</td>
<td>Displays verbose information</td>
</tr>
<tr>
<td>follow</td>
<td>Follows links</td>
</tr>
<tr>
<td>levels=n</td>
<td>Lists n or all directory levels, respectively</td>
</tr>
<tr>
<td>order=...</td>
<td>Sort output by alpha, date, or insert</td>
</tr>
</tbody>
</table>