APPENDIX 2 METHOD DOCUMENTATION

TEXT METHODS

FORMAT_TEXT

Syntax

Object.format_text(< optional argument >, … , < optional argument >);

Description

Displays text in the active output destination(s).

Optional Arguments

data < string | number | character variable | numeric variable >

The data value to display. If the data is numeric and no format has been specified, the data value will be formatted using the BEST. format.

format < string | character variable >

The SAS format to be applied to the data argument.

style < string | character variable >

The style element that contains the collection of style attributes to be applied to the data value. The default style element is USERTEXT.

overrides < string | character variable >

The style attributes to override those defined in the selected style element.

split < string | character variable >

Split character to be applied to the data value. A new line will be started when it reaches the specified split character, and will continue on the next line. The split character itself is not considered part of the data value.

no_clean < number | numeric variable >

just < single character | single character variable >

Horizontal justification for the data value.

Valid values

L	Left justification
C	Center justification
R	Right justification
D	Decimal point justification

vjust < single character | single character variable >

Vertical justification for the data value.

Valid values

T	Top justification
M	Middle justification
B	Bottom justification

Example

```
obj.format_text(data: “Display this text in the active output
destination.”);

obj.format_text(data: “Make this text look like The SAS System
title.”,
                    style: “TitlesandFooters”);

obj.format_text(data: “Make this text red”,
                overrides: “color=red”,
                data: “and this bold”,
                overrides: “font_weight=bold”,
                data: “ and use a 16pt font”,
                overrides: “font_size=16pt”);
```

NOTE

Syntax

Object.note(< optional argument >, … , < optional argument >);

Description

Writes a note to the active output destination(s).

Optional Arguments

data	< string | number | character variable | numeric variable >
	The data value to display. If the data is numeric and no format has been specified the data value will be formatted using the BEST. format.
format	< string | character variable >
	The SAS format to be applied to the data argument.
style	< string | character variable >
	The style element that contains the collection of style attributes to be applied to the data value. The default style element is “note”.
overrides	< string | character variable >
	The style attributes to override those defined in the selected style element.
split	< string | character variable >
	Split character to be applied to the data value. A new line will be started when it reaches the specified split character, and will continue on the next line. The split character itself is not considered part of the data value.
no_clean	< number | numeric variable >
just	< single character | single character variable >
	Horizontal justification for the data value.
	Valid values
	‘L’ Left justification ‘C’ Center justification ‘R’ Right justification ‘D’ Decimal point justification
vjust	< single character | single character variable >

Vertical justification for the data value.

Valid values

'T'	Top justification
'M'	Middle justification
'B'	Bottom justification

Example

```
obj.note(data: "This is the text to display in a note.");
```

TABLE SECTION METHODS

HEAD_START

Syntax

Object.head_start(< optional argument >, … , < optional argument >);

Description

Marks the start of the Table Header Section. The HEAD_START is always used in conjunction with the HEAD_END method. You can also use the TYPE argument with the ROW_START method as an alternate approach.

Optional Arguments

style — < string | character variable >

The style element that contains the collection of style attributes to be applied to the data value. The default style element is "Header".

overrides — < string | character variable >

The style attributes to override those defined in the selected style element.

Example

```
obj.table_start();
  obj.head_start();
    obj.row_start();
     obj.format_cell(data: "A single cell table");
    obj.row_end();
  obj.head_end();
obj.table_end();
```

HEAD_END

Syntax

Object.head_end( );

Description

Marks the end of the Table Header section. The HEAD_END is always used in conjunction with the HEAD_START method.

No Arguments

BODY_START

Syntax

Object.body_start(< optional argument >, … , < optional argument >);

Description

Marks the start of the Table Body(Data) Section. The BODY_START is always used in conjunction with the BODY_END method. You can also use the “TYPE” argument with the ROW_START method as an alternate approach. This is the default section if you just use the ROW_START method.

Optional Arguments

style < string | character variable >

The style element that contains the collection of style attributes to be applied to the data value. The default style element is “Body”.

overrides < string | character variable >

The style attributes to override those defined in the selected style element.

Example

```
obj.table_start();
  obj.body_start();
    obj.row_start();
      obj.format_cell(data: “A single cell table”);
    obj.row_end();
  obj.body_end();
obj.table_end();
```

BODY_END

Syntax

Object.body_end( );

Description

Marks the end of the Table Body(Data) section. The BODY_END is always used in conjunction with the BODY_START method.

No Arguments

FOOT_START

Syntax

Object.foot_start(< optional argument >, … , < optional argument >);

Description

Marks the start of the Table Footer Section. The FOOT_START is always used in conjunction with the FOOT_END method. You can also use the TYPE argument with the ROW_START method as an alternate approach.

Optional Arguments

style < string | character variable >

The style element that contains the collection of style attributes to be applied to the data value. The default style element is “Footer”.

overrides < string | character variable >

The style attributes to override those defined in the selected style element.

Example

```
obj.table_start();
  obj.foot_start();
    obj.row_start();
```

```
      obj.format_cell(data: “A single cell table”);
    obj.row_end();
  obj.foot_end();
obj.table_end();
```

FOOT_END

Syntax

Object.foot_end( );

Description

Marks the end of the Table Footer section. The FOOT_END is always used in conjunction with the FOOT_START method.

No Arguments

TABLE METHODS

TABLE_START

Syntax

Object.table_start(< optional argument >, … , < optional argument >);

Description

The TABLE_START is always used in conjunction with the TABLE _END method.

Optional Arguments

name — The name of the table that will be used in the table of contents (TOC) and DMS Results window.

label — The label of the table that will be used in the table of contents (TOC) and DMS Results window.

style — < string | character variable >

The style element that contains the collection of style attributes to be applied to the data value. The default style element is TABLE.

overrides — < string | character variable >

The style attributes to override those defined in the selected style element.

just — < single character | single character variable >

Horizontal justification for the data value.

Valid values

L	Left justification
C	Center justification
R	Right justification
D	Decimal point justification

vjust — < single character | single character variable >

Vertical justification for the data value.

Valid values

T	Top justification
M	Middle justification
B	Bottom justification

top_space — < number | numeric variable >

The number of blank lines to insert before the table starts.

Example

```
obj.table_start();
  obj.row_start();
    obj.format_cell(data: "A single cell table");
  obj.row_end();
obj.table_end();
```

TABLE_END

Syntax

Object.table_end( );

Description

The TABLE_END is always used in conjunction with the TABLE_START method.

No Arguments

Example

```
obj.table_start();
  obj.row_start();
    obj.format_cell(data: "A single cell table");
  obj.row_end();
obj.table_end();
```

ROW_START

Syntax

Object.row_start(< optional argument >, … , < optional argument >);

Description

The ROW_END is always used in conjunction with the ROW_START method.

Optional Arguments

type — Correspond to the table sections. This is just an alternative to having to use the HEAD_START and HEAD_END method calls. The default type is BODY.

Valid values

H	Header Section
D	Body(data) Section
B	Body(data) Section
F	Footer Section

style — < string | character variable >

The style element that contains the collection of style attributes to be applied to the data value. The default style element depends on the row type.

overrides — < string | character variable >

The style attributes to override those defined in the selected style element.

Row — The row index. The table keeps track of its current row index so this allows you to skip blank rows. Once you have skipped a row, you cannot index back to a previous row.

Examples

```
obj.table_start();
  obj.row_start();
    obj.format_cell(data: "A single cell table");
  obj.row_end();
obj.table_end();
```

ROW_END

Syntax

Object.row_end( );

Description

The ROW_END is always used in conjunction with the ROW_START method.

No Arguments

Example

```
obj.table_start();
  obj.row_start();
    obj.format_cell(data: “A single cell table”);
  obj.row_end();
obj.table_end();
```

CELL_START

Syntax

Object.cell_start(< optional argument >, … , < optional argument >);

Description

The CELL_END is always used in conjunction with the CELL_START method.

Optional Arguments

data — < string | number | character variable | numeric variable >

The data value to display. If the data is numeric and no format has been specified, the data value will be formatted using the BEST. format.

format — < string | character variable >

The SAS format to be applied to the data argument.

style — < string | character variable >

The style element that contains the collection of style attributes to be applied to the data value. The default style element is TEXT.

overrides — < string | character variable >

The style attributes to override those defined in the selected style element.

split — < string | character variable >

Split character to be applied to the data value. A new line will be started when it reaches the specified split character, and will continue on the next line. The split character itself is not considered part of the data value.

no_clean — < number | numeric variable >

Column — < number | numeric variable >

Column_span — < number | numeric variable >

Row_span — < number | numeric variable >

Inhibit — Tells aspects of the cells that may be inhibited. This option can be honored only for certain destinations; in particular, HTML does not currently support it.

Valid values

- T — Do not draw the top border. Note that some destinations will have already drawn a rule at the bottom of the previous row, so this one may not be effective.
- B — Do not draw the bottom border of this cell.
- L — Do not draw the left border. Ineffective if the destination already drew that rule on the right of the previous cell.
- R — Do not draw the right border of this cell.

X	Do not draw the contents of the cell, just the background. Usually desirable on one of the two cells which are using the “B” or ”R”.

Example

```
obj.table_start();
  obj.row_start();
    obj.cell_start()
      obj.format_text(data: “A single cell table”);
    obj.cell_end();
  obj.row_end();
obj.table_end();
```

CELL_END

Syntax

Object.cell_end( );

Description

The CELL_END is always used in conjunction with the CELL_START method.

No Arguments

Example

```
obj.table_start();
  obj.row_start();
    obj.cell_start()
      obj.format_text(data: “A single cell table”);
    obj.cell_end();
  obj.row_end();
obj.table_end();
```

FORMAT_CELL

Syntax

Object.format_cell(< optional argument >, … , < optional argument >);

Description

Optional Arguments

data	< string | number | character variable | numeric variable >
	The data value to display. If the data is numeric and no format has been specified the data value will be formatted using the BEST. format.
format	< string | character variable >
	The SAS format to be applied to the data argument.
style	< string | character variable >
	The style element that contains the collection of style attributes to be applied to the data value. The default style element is TEXT.
overrides	< string | character variable >
	The style attributes to override those defined in the selected style element.
split	< string | character variable >
	Split character to be applied to the data value. A new line will be started when it reaches the specified split character, and will continue on the next line. The split character itself is not considered part of the data value.

no_clean	< number | numeric variable >
Column	< number | numeric variable >
Column_span	< number | numeric variable >
Row_span	< number | numeric variable >
Inhibit	Tells aspects of the cells that may be inhibited. This option can be honored only for certain destinations; in particular, HTML does not currently support it.

Valid values

T	Do not draw the top border. Note that some destinations will have already drawn a rule at the bottom of the previous row, so this one may not be effective.
B	Do not draw the bottom border of this cell.
L	Do not draw the left border. Ineffective if the destination already drew that rule on the right of the previous cell.
R	Do not draw the right border of this cell.
X	Do not draw the contents of the cell, just the background. Usually desirable on one of the two cells which are using the ”B or ”R”.

Example

```
obj.table_start();

  /* Row 1 */
  obj.row_start();
    obj.format_cell(data: “Cell 1”);
    obj.format_cell(data: “Cell 2”);
  obj.row_end();

  /* Row 2 */
  obj.row_start();
    obj.format_cell(data: “Cell 1”);
    obj.format_cell(data: “Cell 2”);
  obj.row_end();

obj.table_end();


obj.table_start();

  /* Row 1 will span multiple cells */
  obj.row_start();
    obj.format_cell(data: “This is a spanning cell”,
                    column_span: 2,
                    style: “Header”);
  obj.row_end();

  /* Row 2 has separate cells */
  obj.row_start();
    obj.format_cell(data: “Cell 1”);
    obj.format_cell(data: “Cell 2”);
  obj.row_end();
obj.table_end();
```

PAGE METHODS

PAGE

Syntax

Object.page( );

Description

Forces a page eject or configures the page.

No Arguments

Example

```
obj.page();
```

TITLE

Syntax

Object.title(< optional argument >, … , < optional argument >);

Description

Adds a new page title to the system.

Optional Arguments

text	< string | character variable >
	The text value to insert into the title system processing.
style	< string | character variable >
	The style element that contains the collection of style attributes to be applied to the data value. The default style element is TITLE.
Start	The title index to begin at. Valid numeric range 1–10.
Clear	Clear the cashed title.

Example

```
obj.title(text: "Here is a new title");

/* Clear "The SAS System" default title */
obj.title(start: 1,
          clear: 1);
```

FOOTNOTE

Syntax

Object.footnote(< optional argument >, … , < optional argument >);

Description

Adds a new page footnote to the system.

Optional Arguments

text	< string | character variable >
	The text value to insert into the footnote system processing.
style	< string | character variable >

	The style element that contains the collection of style attributes to be applied to the data value. The default style element is FOOTNOTE.
Start	The footnote index to begin at. Valid numeric range 1–10.
Clear	Clear the cashed footnote.

Example

```
obj.footnote(text: "Here is a new footnote");

/* Clears the previous footnote */
obj.footnote(start: 1,
             clear: 1);
```

GRIDDED LAYOUT METHODS

LAYOUT_GRIDDED

Syntax

Object.layout_gridded(< optional argument >, … , < optional argument >);

Description

Creates a new gridded layout.

Optional Arguments

x	< dimension unit >
	Horizontal position of the LAYOUT, which will extend to the right of this position for WIDTH. If omitted, it defaults to 0.
y	< dimension unit >
	Vertical position of the LAYOUT, which will extend down from this position for HEIGHT. If omitted, it defaults to the current vertical position on the page.
width	< dimension unit >
	Horizontal width of the LAYOUT. If omitted, it defaults to the maximum horizontal space needed to display all regions.
height	< dimension unit >
	Vertical height of the LAYOUT. If omitted, it defaults to the maximum vertical space needed to display all regions.
columns	< number | numeric variable >
	Fixed number of columns in the gridded LAYOUT. If omitted, it defaults to 1 column.
column_widths	< dimension unit >
	Width of each column specified. This is a space-delimited list of horizontal sizes that correspond to each column. The number of horizontal sizes must match the number of columns specified, or else a warning will be produced, and the option will be ignored.
column_gutter	< dimension unit >
	Horizontal space between each column. If omitted, it defaults to the CELL_SPACING style attribute.

rows < number | numeric variable >

Fixed number of rows in the gridded LAYOUT. If omitted, it defaults to the maximum number of regions created in the vertical direction. This option should be used very sparingly.

row_heights < dimension unit >

Height of each row specified. This is a space-delimited list of vertical sizes that correspond to each row. The number of vertical sizes must match the number of rows specified or else a warning will be produced, and the option will be ignored.

row_gutter < dimension unit >

Vertical space between each row. If omitted, it defaults to the CELL_SPACING style attribute.

style < string | character variable >

The style element that contains the collection of style attributes to be applied to the data value. The default style element is TEXT.

overrides < string | character variable >

The style attributes to override those defined in the selected style element.

Example:

```
obj.layout_gridded(colums: 1);
  obj.region();
    obj.format_text(data: “Here is a some text for a region.”);
obj.layout_end();
```

REGION

Syntax

Object.region(< optional argument >, … , < optional argument >);

Description

Create a region that will contain some output.

Optional Arguments

width Horizontal width of the REGION, and is restricted by the LAYOUT dimensions. If omitted, it defaults to the maximum horizontal space needed to display the output contained in the REGION. The sum of all region widths cannot exceed the LAYOUT horizontal dimension.

height Vertical height of the REGION, and is restricted by the LAYOUT dimensions. If omitted, it defaults to the maximum vertical spaced needed to display the output contained in the REGION. The sum of all region heights cannot exceed the LAYOUT vertical dimension. This option should be used very sparingly.

column Allows you to specify the current grid column position in the gridded layout. This is generally useful only when you want to skip regions in the gridded layout and should be used very sparingly. The gridded layout automatically tracks the current grid column position and will be incremented for every region statement. Once you have skipped a grid column, you cannot go back to them. Random access of grid rows and columns is not supported.

column_span Allows you to specify the number of grid columns that the region will occupy. It simply allows you to combine adjacent grid columns in

gridded layout. Default is 1.

row — Allows you to specify the current grid row position in the gridded layout. This is generally only useful when you want to skip regions in the gridded layout, and should be used very sparingly. The gridded layout automatically tracks the current row position, and will be incremented for every region statement. Once you have skipped rows, you cannot go back to them. Random access to row and columns is not supported.

row_span — Allows you to specify the number of grid rows that the region will occupy. It simply allows you to combine adjacent grid rows in gridded layout. Default is 1.

style — The style element that contains the collection of style attributes to be applied to the data value. The default style element is TEXT.

overrides — The style attributes to override those defined in the selected style element.

Example:

```
obj.layout_gridded(colums: 1);
  obj.region();
    obj.format_text(data: "Here is a some text for a region.");
obj.layout_end();
```

LAYOUT_END

Syntax

Object.layout_end( );

Description

Close the active layout.

No Arguments

Example

```
obj.layout_end();
```

ABSOLUTE LAYOUT METHODS

LAYOUT_ABSOLUTE

Syntax

Object.layout_absolute(< optional argument >, … , < optional argument >);

Description

Create an absolute layout container.

Optional Arguments

x — < dimension unit >

Horizontal position of the LAYOUT, which will extend to the right of this position for WIDTH. If omitted, it defaults to 0.

y — < dimension unit >

Vertical position of the LAYOUT, which will extend down from this

position for HEIGHT. If omitted, it defaults to the current vertical position on the page.

width < dimension unit >

Horizontal width of the LAYOUT. If omitted, it defaults to the maximum horizontal spaced needed to display all regions.

height < dimension unit >

Vertical height of the LAYOUT. If omitted, it defaults to the maximum vertical spaced needed to display all regions.

style < string | character variable >

The style element that contains the collection of style attributes to be applied to the data value. The default style element is TEXT.

overrides < string | character variable >

The style attributes to override those defined in the selected style element.

Example

```
obj.layout_absolute(x: “4in”,
                    y: “3in”,
                    width: “3in”,
                   height: “1in”);
  obj.region();
    obj.format_text(data: “Some text for a region.”);
obj.layout_end();
```

REGION

Syntax

Object.region(< optional argument >, … , < optional argument >);

Description

Optional Arguments

width Horizontal width of the REGION, and is restricted by the LAYOUT dimensions. If omitted, it defaults to the maximum horizontal spaced needed to display the output contained in the REGION. The sum of all region widths cannot exceed the LAYOUT horizontal dimension.

height Vertical height of the REGION, and is restricted by the LAYOUT dimensions. If omitted, it defaults to the maximum vertical spaced needed to display the output contained in the REGION. The sum of all region heights cannot exceed the LAYOUT vertical dimension. This option should be used very sparingly.

style The style element that contains the collection of style attributes to be applied to the data value. The default style element is TEXT.

overrides The style attributes to override those defined in the selected style element.

Example

```
obj.layout_absolute(x: “4in”,
                    y: “3in”,
                    width: “3in”,
```

```
                        height: "1in");
  obj.region();
    obj.format_text(data: "Some text for a region.");
obj.layout_end();
```

LAYOUT_END

Syntax

Object.layout_end( );

Description

No Arguments

Example

```
obj.layout_end();
```

MISCELLANEOUS METHODS

OPEN_DIR

Syntax

Object.open_dir(< optional argument >, … , < optional argument >);

Description

Open a table of contents directory. The OPEN_DIR is always used in conjunction with the CLOSE_DIR method.

Optional Arguments

name	< string | character variable >
	The name of the director that will be used in the table of contents (TOC) and DMS Results window.
label	The label of the directory that will be used in the table of contents (TOC) and DMS Results window.
by	< 1 (True) | 0 (False) >
	This is a directory that is related to the unique by value. Special BY-group processing occurs when this option is set.

Example

```
obj.open_dir(name: "Sub Directory");
  obj.format_text(data: "here is some data.");
obj.close_dir();
```

CLOSE_DIR

Syntax

Object.close_dir( );

Description

Close the open directory. The CLOSE_DIR is always used in conjunction with the OPEN_DIR method.

No Arguments

Example

```
obj.open_dir(name: "Sub Directory");
  obj.format_text(data: "here is some data.");
obj.close_dir();
```

LINE

Syntax

Object.line(< optional argument >, … , < optional argument >);

Description

Draws a horizontal rule (line) across the page.

Optional Arguments

size	< dimension unit > The thickness of the line.
style	The style element that contains the collection of style attributes to be applied to the data value. The default style element is TEXT.

Example

```
obj.line();

obj.line(size: "1mm");
```

IMAGE

Syntax

Object.image(< optional argument >, … , < optional argument >);

Description

Insert the image into all open output destinations.

Optional Arguments

file	< string | character variable > The FILEREF or physical file name of the external image to include.
style	The style element that contains the collection of style attributes to be applied to the data value. The default style element is "text".
overrides	< string | character variable > The style attributes to override those defined in the selected style element.

Example

```
obj.image(file: "c:\someimage.jpg");
```

HREF

Syntax

Object.href(< optional argument >, … , < optional argument >);

Description

Create a link to another document.

Optional Arguments

data	< string | number | character variable | numeric variable >
	The data value to display. If the data is numeric and no format has been specified the data value will be formatted using the BEST format.
format	< string | character variable >
	The SAS format to be applied to the data argument.
href	< string | character variable >
	The URL that our data string.
style	< string | character variable >
	The style element that contains the collection of style attributes to be applied to the data value. The default style element is TEXT.
split	< string | character variable >
	Split character to be applied to the data value. A new line will be started when it reaches the specified split character, and will continue on the next line. The split character itself is not considered part of the data value.
no_base	< numeric | numeric variable >

Example

```
obj.href(data: “SAS Home page”,
         file: “http://www.sas.com/index.html”);
```

DELETE

Syntax

Object.delete( );

Description

Delete an instance of the ODSOUT class.

No Arguments