

Applixware Spreadsheets ELF Reference

COPYRIGHT NOTICE ON THE VERSION 6.0 SOFTWARE
©1990 - 2010 Vistasource, Inc. All Rights Reserved.

Vistasource, Inc. prepared the information contained in this document for use by Vistasource personnel, customers, and prospects. Vistasource reserves the right to change the information in this document without prior notice. The contents herein should not be construed as a representation or warranty by Vistasource. Vistasource assumes no responsibility for any errors that may appear in this document.

The Proximity Thesauri ®
©1985 Merriam-Webster Inc.
©1988 Williams Collins Sons & Co. Ltd.
©1989 Van Dale Lexicografie bv. ©1989 Nathan. ©1989 Kruger.
©1989 Zanichelli. ©1989 International Data Education a s.
©1989 C.A. Stromber A B. ©1989 Espasa-Calpe.
©1983-1996. Proximity Technology, Inc.
All Rights Reserved.

The Proximity Linguibase And Hyphenation Systems®
©1983 Merriam-Webster Inc.
©1984, 1985, 1986, 1988, 1990 Williams Collins Sons & Co. Ltd.
©1987, 1989 Van Dale Lexicografie bv. ©1988 Munksgaard International Publishers Ltd.
©1988, 1989 International Data Education a s.
©1983-1996 Proximity Technology, Inc.
All Rights Reserved

The Applixware Graphics Filter Pack contains elements of the Generator Metafile Development Libraries (MDL/G)
©1988-1996 Henderson Software, Inc.
All Rights Reserved

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraphs (c) (1) (ii) of SFARS 252.277-7013, or in FAR 52.227-19, as applicable.

Hardware and software products mentioned herein are used for identification purposes only and may be trademarks of their respective companies.

Applixware is a registered trademark of Vistasource, Inc. Applixware, Applixware Real Time, Applixware Data, and Applixware Builder are trademarks of Vistasource, Inc.

This manual was produced using Applixware.

Printed: June 2010

SS_ADD_WIN@

Creates a new window of the same spreadsheet

Format SS_ADD_WIN@()

Method [this.add_win@](#)

Description Creates a new window based on the same Spreadsheet document. You can create up to three additional windows of the same spreadsheet.

When you create a new window, the name of the original spreadsheet re-displays with a suffix of :1. The new window has a suffix of :2. A second new window has a suffix of :3, and a third new window has a suffix of :4.

You can use multiple windows to view different sections of a large spreadsheet simultaneously. You can move to any part of your spreadsheet independently in each window and each window of your spreadsheet has its own selection.

You can edit your spreadsheet in any of the new windows as you would in the original spreadsheet. All of the windows of a spreadsheet update together.

SS_APPLICATION_DLG@

Creates a new Spreadsheets window

Format taskID = SS_APPLICATION_DLG@([menubarID[, windowlessFlag] [,hooklessFlag])

Arguments menubarID The number of a menu bar to be associated with this window. (This argument is optional.) This number should be a number between 100 and 199.

windowlessFlag

A Boolean value where TRUE indicates that no window will be displayed. FALSE is the default.

hooklessFlag

A Boolean value where TRUE indicates that no hook macro is run at startup time. If the value is FALSE, Spreadsheets runs the hook macro configured in the Spreadsheets preferences.

Description Creates a new Spreadsheets window. This window can use the default menu bar or it can use the menu bar associated with menubarID. The task id for the newly created task is returned.

The optional `menubarID` parameter lets you load a menu bar according to the task that will be performed in the Spreadsheets window. You could even display several versions at the same time by opening several windows, each with separate menubars. (Before you display a custom menubar, you have to load it into memory using [**SET SELECTIONS@**](#).)

The `windowlessFlag` parameter lets you perform automated tasks "in the background," without displaying the Spreadsheets application window. Using this option, two applications can work simultaneously without interrupting each other and without invoking two separate `axmain` processes. This windowless Spreadsheets application becomes a "child" of the window from which it was invoked.

Performing non-interactive tasks without a window conserves computing resources as the window will never have to be displayed.

To perform a windowless task:

- You must suppress all messages that are displayed. Otherwise, the task will hang when it tries to display them.

To suppress information displayed by `INFO_MESSAGE@`, use [**SUPPRESS INFO MESSAGES@**](#).

Suppress error messages with the [**SUPPRESS ERROR MESSAGES@**](#) macro.

- You cannot include any prompts using `PROMPT@`.
- Be sure to explicitly exit a windowless Spreadsheets application after it completes. Otherwise, the task remains in memory until you log out.
- It is best to make a call to [**SELECT WINDOW@**](#) when exiting the windowless application. In this way, you guarantee that the exit command is invoked against the right window.
- Before testing a newly written macro that invokes a windowless application, it is best to first test the macro with all windows displaying in the foreground.

The `hooklessFlag` parameter determines whether a hook macro is run when the Spreadsheet application starts. A hook macro is configured through the Spreadsheet preferences dialog, and runs whenever you start Spreadsheets or open a spreadsheet file.

If you configure a hook macro to run at startup time, and you set `hooklessFlag` to `NULL`, the hook macro runs. If you configure a hook macro to run at startup time, and you set `hooklessFlag` to `TRUE`, the hook macro does not run. This is an optional parameter.

`SS_APPLICATION_DLG@` is called by the * ® Spreadsheets menu option.

SS_BACKSPACE_KEY@

Deletes the character preceding the cursor

Format SS_BACKSPACE_KEY@()

Method [this.backspace_key@](#)

See also [SS_DELETE_KEY@](#)

SS_BACK_RETURN_KEY@

Moves back a cell

Format SS_BACK_RETURN_KEY@()

Method [this.back_return_key@](#)

Description Moves the cell pointer one cell to the left.

See also [SS_DOWN_ARROW_KEY@](#)

[SS_LEFT_ARROW_KEY@](#)

[SS_RETURN_KEY@](#)

[SS_RIGHT_ARROW_KEY@](#)

[SS_UP_ARROW_KEY@](#)

SS_BLANK@

Deletes the contents of the current cell or all selected cells

Format SS_BLANK@()

Method [this.blank@](#)

Description If no cells are selected, SS_BLANK@ deletes the data from the cell in which the cursor currently resides. If cells are selected, SS_BLANK@ deletes the data from all the selected cells. The style settings for the cells are not deleted.

Data deleted by SS_BLANK@ is not saved to the clipboard and can not be recovered.

See also [SS_CLEAR@](#)
[SS_BLANK_RANGE@](#)

SS_BLANK_RANGE@

Deletes information from cells

Format SS_BLANK_RANGE@(ranges)

Method [this.blank_range@](#)(ranges)

Arguments ranges The range of cells from which to delete information. More than one range can be specified.

Description Deletes the contents of all specified cells in the current Spreadsheets document. Unlike [SS_DELETE_RANGE@](#), SS_BLANK_RANGE@ does not delete the style settings for the cells. Only the cell contents are deleted. The deleted information is not saved in the clipboard.

SS_BLANK_RANGE@ is called by the Edit ® Blank menu option.

See also [SS_BLANK@](#)

SS_BOLD@

Makes selected text bold

Format SS_BOLD@()

Method [this.bold@](#)

Description Makes selected text bold in a Spreadsheet document. If no text is selected, makes the current cell bold. SS_BOLD@ is called by the Style ® Bold menu option.

See also [SS_ITALICS@](#)
[SS_UNDERLINE@](#)

SS_BOTTOM_SECTION@

Moves the cursor down to the next cell containing data

Format SS_BOTTOM_SECTION@()

Method [this.bottom_section@](#)

Description Moves the cursor from its current position to the next cell down that contains data. If no cells containing data are found below the current cursor position, the cursor is moved to the last cell in the column. SS_BOTTOM_SECTION@ is called by the Keys ® Next data down menu option.

See also [SS_TOP_SECTION@](#)

SS_CALC@

Sets the calculation mode for a spreadsheet.

NOTE: This macro is obsolete. Use [SS_SET_CALC_OPTIONS@](#).

Format SS_CALC@(freq, type, count, autoChartFlag, calcInterval, minimalRecalcFlag)

Method [this.calc@\(freq, type, count, autoChartFlag, calcInterval, minimalRecalcFlag\)](#)

Arguments

freq	The frequency of calculation. freq can be: 0 manual calculation 1 automatic calculation
type	The type of calculation performed. type can be: 0 natural calculation starts in the upper left corner of the spreadsheet 1 row calculation is performed in row order 2 column calculation is performed in column order
count	Indicates the number of calculation iterations to be performed. count can be a number from 1 to 10. This argument only applies when row or column calculations are specified, in which case multiple iterations may be required to resolve forward references in formulas. If type is set to 0 (natural calculation), count is ignored.

autoChartFlag

A Boolean value where TRUE indicates that a chart will be redrawn based upon the calculations made.

calcInterval A numeric value indicating the number of seconds between recalculations. The default interval is 5 seconds.

minimalRecalcFlag

A Boolean value where TRUE indicates that only obsolete cells are recalculated. FALSE indicates that all cells are recalculated.

Description If automatic calculation is set, cells are automatically recalculated when new data is typed. If manual calculation is set, cells are not recalculated until the Recalculation menu option is chosen. When in manual calculation mode, all cells that need to be recalculated display OBSOLETE. Spreadsheet defaults are for automatic, natural calculation.

Manual calculation is provided for compatibility with VisiCalc spreadsheets. Manual recalculation degrades Applixware Spreadsheets performance and should not be used unless necessary.

See also [SS_RECALC@](#)

SS_CELL_FILE@

Inserts a document into a spreadsheet cell

Format SS_CELL_FILE@()

Method [this.cell_file@](#)

Description Inserts a document into a spreadsheet cell.

SS_CELL_JUSTIFY@

Sets the justification for cells in the range

Format SS_CELL_JUSTIFY@(ranges, type, wrapTextFlag[, vertAlign])

Method [this.cell_justify@\(ranges, type, wrapTextFlag\[, vertAlign\]\)](#)

Arguments

ranges	The ranges whose justification is being set.
type	0 no justification SSC#JUST_LEFT

left justification
SSC#JUST_RIGHT
right justification
SSC#JUST_CENTER
center justification
SSC#JUST_REPEAT
repeat

wrapTextFlag

A Boolean value where TRUE means that the text will be wrapped within the cell.

vertAlign

A number indicating how the text is aligned vertically in the cell, as follows:

0 Default
1 Top
2 Center
3 Bottom

Description Sets the justification and wrapping state for information contained within one or more ranges.

See also [SS JUSTIFY ?@](#)
[SS JUSTIFY SELECTED@](#)

SS_CHANGE_NAMED_RANGE@

Changes range information

Format SS_CHANGE_NAMED_RANGE@(oldName, newName, range[, refCell])

Method [this.change_named_range@\(oldName, newName, range\[, refCell \]\)](#)

Arguments

oldName	The range whose data is being changed.
newName	The new name of the range.
range	The range specification for the range being changed.
refCell	If this argument exists, you are entering the address of a reference cell for a relative range.

Description Changes range information by altering the range's name and cell specifications. If you are just changing the range's extent, set newName equal to oldName. Similarly, if you want to change a range's name without changing its extent, set range to the existing range extent.

SS_CHART_CHANGE_CHART_TYPE@

Changes a chart's type and then redraws chart

Format SS_CHART_CHANGE_CHART_TYPE@(chartName, chartType)

Method [this.chart_change_chart_type@](#)(chartName, chartType)

Arguments

chartName	The string name of a chart
chartType	A string indicating the new type of chart to display. This string consists of a name followed by a number, as follows:

Name	Number Range	Examples
Line	1 - 11	Line1, Line11
Column	1 - 6	Column1, Column6
Bar	1 - 6	Bar1, Bar6
Pie	1 - 6	Pie1, Pie6
Strata	1 - 3	Strata1, Strata3
3DColumn	1 - 4	3DColumn1, 3DColumn4
3DLine	1 - 4	3DLine4
3DBar	1 - 7	3DBar1, 3DBar7
3DPie	1 - 6	3DPie1, 3DPie6
3DStrata	1 - 3	3DStrata1, 3DStrata3
Histogram	1 - 2	Histogram1
Bubble	1	Bubble1
Combination	1 - 6	Combination1 Combination6
XY	1 - 12	XY1, XY12

Description Changes the type of the target chart, and re-draws the chart on the screen. The types in the table can be seen in the Charts ® Create ® Step-by-Step dialog.

SS_CHART_CREATE@

Creates a chart data structure in memory

Format SS_CHART_CREATE@(chartName, format SS_OBJECT_LOC@ location)

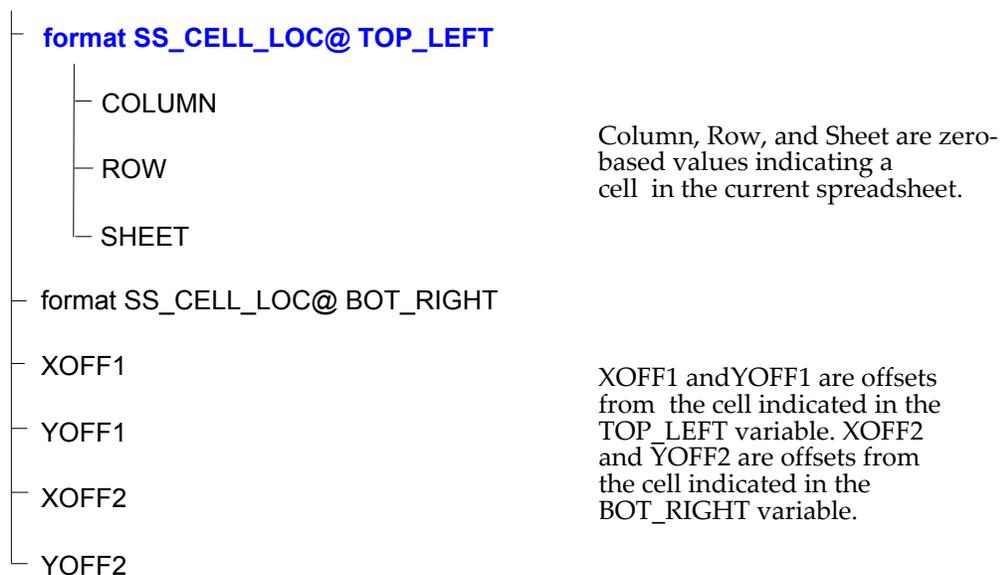
Method this.chart_create@(chartName, format SS_OBJECT_LOC@ location)

Arguments chartName A string name. This string becomes the name of the chart object.
location An array of format SS_OBJECT_LOC@ that defines the location of the chart within the spreadsheet.

Description Creates a chart structure in memory. You must use [SS_CHART_FORMAT@](#) to establish the type and attributes of the chart, and the macro [SS_CHART_DRAW@](#) to display the chart in the spreadsheet.

The structure of SS_OBJECT_LOC@ is shown in the following diagram.

SS_OBJECT_LOC@



SS_CHART_DEFINE_AXIS_RANGE@

Sets the range that contains the axis definition

Format SS_CHART_DEFINE_AXIS_RANGE@(chartName, axisName, range)

Method [this.chart_define_axis_range@\(chartName, axisName, range\)](#)

Arguments

chartName	The name of the chart.
axisName	The name of one of the chart's axes. This name takes the following form: axis x1 axis x2 ... axis y1 axis y2 ...
range	The range containing the axis' tick mark information.

Description Sets the range of cells that contains the information that will contain axis information.

See also [SS_CHART_DEFINE_AXIS_TITLE@](#)
[SS_CHART_DEFINE_DATA_GROUP@](#)
[SS_CHART_DEFINE_FOOTER@](#)
[SS_CHART_DEFINE_LEGEND_LABELS@](#)
[SS_CHART_DEFINE_LEGEND_TITLE@](#)
[SS_CHART_DEFINE_SUBTITLE@](#)
[SS_CHART_DEFINE_TITLE@](#)

SS_CHART_DEFINE_AXIS_TITLE@

Sets the cell that defines the axis title

Format SS_CHART_DEFINE_AXIS_TITLE@(chartName, axisName, cell)

Method [this.chart_define_axis_title@\(chartName, axisName, cell\)](#)

Arguments

chartName	The name of the chart.
axisName	The name of one of the chart's axes. This name takes the following form:

axis x1
axis x2
...
axis y1
axis y2
...
cell The cell containing the title string.

Description Names the cell that contains the string that will be used when an axis title is displayed.

See also [SS_CHART_DEFINE_AXIS_RANGE@](#)
[SS_CHART_DEFINE_DATA_GROUP@](#)
[SS_CHART_DEFINE_FOOTER@](#)
[SS_CHART_DEFINE_LEGEND_LABELS@](#)
[SS_CHART_DEFINE_LEGEND_TITLE@](#)
[SS_CHART_DEFINE_SUBTITLE@](#)
[SS_CHART_DEFINE_TITLE@](#)

SS_CHART_DEFINE_DATA_GROUP@

Sets the chart type and the range that contains the group's data

Format SS_CHART_DEFINE_DATA_GROUP@(chartName, groupName, rangeArray, chartType)

Method [this.chart_define_data_group@\(chartName, groupName, rangeArray, chartType\)](#)

Arguments

chartName	The name of the chart.
groupName	The name of the group being created. If this argument is NULL or is not a string, a name is created for you. A groupName created by Spreadsheets is in the following form: data 0 data 1 data 2 However, the value following data can be any user-defined string
rangeArray	The contents of this array differ according to the type of graph. For example, for a val-val chart, one range would contain the X values and the other would contain the Y values. For a HiLo chart, four ranges are necessary. For line charts, each range has the values for one line.

chartType One of the following:

percent
cat-val
val-val
histogram
strata

By setting the chart types of groups differently, you can show more than one charting style within one chart.

Description Defines the range in the spreadsheet that contains information for one of the chart's groups. The way in which this information is displayed is defined using the chartType argument.

See also [SS_CHART_DEFINE_AXIS_RANGE@](#)
[SS_CHART_DEFINE_AXIS_TITLE@](#)
[SS_CHART_DEFINE_FOOTER@](#)
[SS_CHART_DEFINE_LEGEND_LABELS@](#)
[SS_CHART_DEFINE_LEGEND_TITLE@](#)
[SS_CHART_DEFINE_SUBTITLE@](#)
[SS_CHART_DEFINE_TITLE@](#)

SS_CHART_DEFINE_FOOTER@

Sets the cell that contains the chart's footer

Format SS_CHART_DEFINE_FOOTER@(chartName, cell)

Method [this.chart_define_footer@\(chartName, cell\)](#)

Arguments chartName The name of the chart.
cell A cell address.

Description Names the cell that contains the string that is used when the chart's footer title is displayed.

See also [SS_CHART_DEFINE_AXIS_RANGE@](#)
[SS_CHART_DEFINE_AXIS_TITLE@](#)
[SS_CHART_DEFINE_DATA_GROUP@](#)
[SS_CHART_DEFINE_LEGEND_LABELS@](#)

[SS_CHART_DEFINE_LEGEND_TITLE@](#)

[SS_CHART_DEFINE_SUBTITLE@](#)

[SS_CHART_DEFINE_TITLE@](#)

SS_CHART_DEFINE_LEGEND_LABELS@

Sets legend information

Format SS_CHART_DEFINE_LEGEND_LABELS@(chartName, format ss_legend_descr@ legendArray)

Method this.chart_define_legend_labels@(chartName, format ss_legend_descr@ legendArray)

Arguments chartName The name of the chart.
legendArray An array of ss_legend_descr@ data where each array element defines one entry in the legend.

Description Sets a chart's legend information as well as associating the data group with a legend and the legend information.

The definition of ss_legend_descr@ is as follows:

format ss_legend_descr@
group_name, ' The name of the data group
cell ' The cell containing the legend

See also [SS_CHART_DEFINE_AXIS_RANGE@](#)

[SS_CHART_DEFINE_AXIS_TITLE@](#)

[SS_CHART_DEFINE_DATA_GROUP@](#)

[SS_CHART_DEFINE_FOOTER@](#)

[SS_CHART_DEFINE_LEGEND_TITLE@](#)

[SS_CHART_DEFINE_SUBTITLE@](#)

[SS_CHART_DEFINE_TITLE@](#)

SS_CHART_DEFINE_LEGEND_TITLE@

Sets the cell containing the legend's title

Format SS_CHART_DEFINE_LEGEND_TITLE@(chartName, cell)

Method `this.chart_define_legend_title@(chartName, cell)`

Arguments `chartName` The name of the chart.
`cell` A cell address.

Description Names the cell that contains the string that will be used when the chart's legend is displayed.

See also [SS_CHART_DEFINE_AXIS_RANGE@](#)
[SS_CHART_DEFINE_AXIS_TITLE@](#)
[SS_CHART_DEFINE_DATA_GROUP@](#)
[SS_CHART_DEFINE_FOOTER@](#)
[SS_CHART_DEFINE_LEGEND_LABELS@](#)
[SS_CHART_DEFINE_SUBTITLE@](#)
[SS_CHART_DEFINE_TITLE@](#)

SS_CHART_DEFINE_SUBTITLE@

Sets the cell containing the chart's subtitle

Format `SS_CHART_DEFINE_SUBTITLE@(chartName, cell)`

Method `this.chart_define_subtitle@(chartName, cell)`

Arguments `chartName` The name of the chart.
`cell` A cell address.

Description Names the cell that contains the string that will be used when the chart's subtitle is displayed.

See also [SS_CHART_DEFINE_AXIS_RANGE@](#)
[SS_CHART_DEFINE_AXIS_TITLE@](#)
[SS_CHART_DEFINE_DATA_GROUP@](#)
[SS_CHART_DEFINE_FOOTER@](#)
[SS_CHART_DEFINE_LEGEND_LABELS@](#)
[SS_CHART_DEFINE_LEGEND_TITLE@](#)
[SS_CHART_DEFINE_TITLE@](#)

SS_CHART_DEFINE_TITLE@

Sets the cell containing the chart's title

Format SS_CHART_DEFINE_TITLE@(chartName, cell)

Method [this.chart_define_title@\(chartName, cell\)](#)

Arguments chartName The name of the chart.
cell A cell address.

Description Names the cell that contains the string that will be used when the chart's title is displayed.

See also [SS_CHART_DEFINE_AXIS_RANGE@](#)
[SS_CHART_DEFINE_AXIS_TITLE@](#)
[SS_CHART_DEFINE_DATA_GROUP@](#)
[SS_CHART_DEFINE_FOOTER@](#)
[SS_CHART_DEFINE_LEGEND_LABELS@](#)
[SS_CHART_DEFINE_LEGEND_TITLE@](#)
[SS_CHART_DEFINE_SUBTITLE@](#)

SS_CHART_DESTROY@

Deletes a chart object from the Spreadsheets document

Format SS_CHART_DESTROY@(chartName)

Method [this.chart_destroy@\(chartName\)](#)

Arguments chartName The string name of a chart

Description Deletes a chart from the current spreadsheet document.

SS_CHART_DRAW@

Draws (or redraws) a chart

Format SS_CHART_DRAW@(chartName)

Method [this.chart_draw@\(chartName\)](#)

Arguments chartName The string name of a chart

Description Displays a chart on the screen. The type of chart, its display attributes, and its location must have been previously defined using the macros [SS_CHART_CREATE@](#) and [SS_CHART_FORMAT@](#).

SS_CHART_FORMAT@

Formats a chart

Format SS_CHART_FORMAT@(chartName, type, range, format chart_step_order@ rangeOrder, format chart_step_titles@ titles, showLegend)

Method [this.chart_format@\(chartName, type, range, format chart_step_order@ rangeOrder, format chart_step_titles@ titles, showLegend\)](#)

Arguments chartName The string name of a chart in the current spreadsheet.
type A string indicating the type of chart to be formatted. This string consists of a name followed by a number, as follows:

Name	Number Range	Examples
Line	1 - 11	Line1, Line11
Column	1 - 6	Column1, Column6
Bar	1 - 6	Bar1, Bar6
Pie	1 - 6	Pie1, Pie6
Strata	1 - 3	Strata1, Strata3
3DColumn	1 - 4	3DColumn1, 3DColumn4
3DLine	1 - 4	3DLine4
3DBar	1 - 7	3DBar1, 3DBar7

3DPie	1 - 6	3DPie1, 3DPie6
3DStrata	1 - 3	3DStrata1, 3DStrata3
Histogram	1 - 2	Histogram1
Bubble	1	Bubble1
Combination	1 - 6	Combination1 Combination6
XY	1 - 12	XY1, XY12

- range A range of spreadsheet cells that contain data to be displayed in the chart
- rangeOrder A chart_step_order@ format that indicates the order of information displayed in the chart.
- titles An array of format chart_step_titles@ containing the title, subtitle, Axis labels and footnotes of the chart.
- showLegend A Boolean indicating whether or not the legend should be displayed in the chart

Description Formats a chart. This macro must be preceded by the macro [SS CHART CREATE@](#). To display the formatted chart, you use the macro [SS CHART DRAW@](#).

The formats chart_step_order@ and chart_step_titles@ are defined in the ELF include file spsheet_.am. The titles argument should be a format of arrays. For example, you could use the following code fragment to load the chart_step_titles@ format:

```
titles.title[0] = "Title"
titles.subtitle[0] = "Subtitle"
titles.footer[0] = "Footer"
titles.x_axis[0] = "X-Axis"
titles.y_axis[0] = "Y-Axis"
```

The rangeOrder argument contains an array of format chart_step_order@. That format is shown in the following diagram.

chart_step_order@

CH_ORDER	CH_ORDER is a string. If this string is "Row", the data is displayed in the chart row by row. If this string is "Column" the data is displayed in the chart column by column.
USE_ROWS	
USE_COLUMNS	USE_ROWS and USE_COLUMNS are Booleans.

If USE_ROWS is set to 1, then the first row of the chart is used for the Category Axis labels.

If USE_COLUMNS is set to 1, then the first column of the chart is used for the Chart legend.

SS_CHART_SELECT@

Selects a chart object

Format SS_CHART_SELECT@(chartName)

Method this.chart_select@(chartName)

Arguments chartName The string name of a chart in the current spreadsheet.

Description Selects the named chart object. The name of the chart is case-sensitive.

SS_CHART_SET_3D@

Defines general 3D information: pitch, yaw, projection, and depth

Format SS_CHART_SET_3D@(chartName, format chart_3D_effect@ 3D)

Method this.chart_set_3d@(chartName, format chart_3D_effect@ 3D)

Arguments chartName The name of a chart.

3D A structure containing 3D definitions.

Description Defines some general 3D characteristics of the chart, as follows:

format chart_3D_effect@
enabled,
ya2,
pitch,

projection,
depth

SS_CHART_SET_ATTR@

Sets graphic attributes for a chart

Format SS_CHART_SET_ATTR@(chartName, range, format gr_attribute@ info, noDrawFlag)

Method [this.chart_set_attr@](#)(chartName, range, format gr_attribute@ info, noDrawFlag)

Arguments

chartName	The name of the chart.
range	A range containing the information. The kinds of elements that can be set include: default minor horizontal grid major horizontal grid major vertical grid minor vertical grid line, axis x<number> line, axis y<number> label, axis x<number> label, axis y<number> tik label, axis x<number> tik label, axis y<number> data label, data title subtitle footer legend box legend title legend labels
info	The attributes being redefined.
noDrawFlag	A Boolean value where TRUE indicates that the chart should not be re-drawn when the new attributes are applied.

Description Sets the graphic display attributes for an element in the chart. For more information, see [GR_CHART_GET_ATTR@](#).

If you are not setting (or using) an attribute within the format, set the attribute's value to NULL.

SS_CHART_SET_AXIS_LABELS@

Sets axis label information and parameters

Format SS_CHART_SET_AXIS_LABELS@(chartName, range, format chart_axis_labels@ info)

Method this.chart_set_axis_labels@(chartName, range, format chart_axis_labels@ info)

Arguments

chartName	The name of the chart.
addr	The range containing the name of one of the chart's axes. This name takes the following form: axis x1 axis x2 ... axis y1 axis y2 ...
info	The axis label information.

Description Defines the properties and attributes of an axis label. This information is defined as a chart_axis_labels@ format whose definition is as follows:

```
format chart_axis_labels@
    type,          'category, value, log10
    tight_label,
    label,
    label_x_margin,
    label_y_margin,
    label_alignment,
    format chart_number_format@ tik_format,
    tight_tiks,
    tik_labels,
    tik_margin
    tik_alignment,
    tik_filter,
    tik_flags,
    hidden
```

The definition of chart_number_format@ is as follows:

```
format chart_number_format@
    style,
```

aux_style_info,
 units,
 radix_places,
 prefix,
 suffix,
 am_str,
 prm_str,
 true_str,
 false_str,
 thousands_seperator,
 decimal_separator,
 display_format_errors,
 trim_string

SS_CHART_SET_AXIS_LINE@

Sets axis line information

Format SS_CHART_SET_AXIS_LINE@(chartName, range, format chart_axis@ info)

Method [this.chart_set_axis_line@](#)(chartName, range, format chart_axis@ info)

Arguments

chartName	The name of the chart.
range	The place where the name of the axis being set is stored. This name takes the following form: axis x1 axis x2 ... axis y1 axis y2 ...
info	Information describing one of a chart's axis.

Description Sets the properties and attributes of one of a chart's axis lines. The definition of the format used when invoking this macro is as follows:

```

format chart_axis@
    type,          'category, value, log10
    auto_max,
    max_value,
    auto_min,
    min_value,
    auto_bas,
  
```

bas_val,
position,
floating_value,
minor_tik_type,
minor_tik_size,
auto_n_minor_tiks,
n_minor_tiks,
major_tik_type,
major_tik_size,
auto_n_major_tiks,
n_major_tiks,
bar_margin,
bar_overlap,
use_tik_attriutes,
behind_data,
hidden

SS_CHART_SET_DECORATIONS@

Sets a chart's border and grid attributes

Format SS_CHART_SET_DECORATIONS@(chartName, format chart_decorations@ info)

Method [this.chart_set_decorations@](#)(chartName, format chart_decorations@ info)

Arguments chartName The name of a chart.

info A set of TRUE/FALSE values indicating if one of the eight kinds of chart *decoration* is drawn.

Description Sets the following format, which contains 8 Boolean chart properties:

format chart_decorations@
close_top,
close_bottom,
close_left,
close_right,
major_h_grids, ' horizontal
minor_h_grids,
major_v_grids, ' vertical
minor_v_grids

SS_CHART_SET_DEFAULT_PROFILE@

Sets the Spreadsheets profile for the default chart type

Format SS_CHART_SET_DEFAULT_PROFILE@(chart_default)

Arguments chart_default A string. This string must contain one of the following values:

Column	3DColumn	combination
Bar	3DBar	
Strata	3DStrata	
Line	3DLine	
Pie	3DPie	
Histogram	XY	

Description Sets the default chart type in the Spreadsheet.

SS_CHART_SET_GROUP@

Sets a data group

Format SS_CHART_SET_GROUP@(chartName, range, format chart_group@ info)

Method [this.chart_set_group@](#)(chartName, range, format chart_group@ info)

Arguments chartName The name of a chart.

range The cell containing the name of the group being created. If this argument is NULL or is not a string, a name is created for you. A groupName created by Graphics is in the following form:

data 0
data 1
data 2
...

However, the value following data can be any user-defined string.

The data for the group.

Description Defines how a group is drawn. The definition of how it is drawn is contained within in-Grp.

format chart_group@

type,	0	line
	1	curve
	2	bar
	3	league bar
	4	custom bar
	5	bubble
	6	hilo1
	7	strata
	8	pie
	9	hilo2
	10	stacked bar
	11	stacked custom bar
	12	stacked strata
stack_id,		
custom_part_name,		
x_axis,		parent x axis
y_axis,		parent y axis
z_axis,		parent z axis
label_x_offset,		label's x offset
label_y_offset,		label's y offset
label_type,	0	none
	1	value
	2	percent
	3	string
format chart_number_format@		label_format,
format arrayof chart_datum@		data,
legend_string,		An array or a string
label_alignment,		left
		center
		right

SS_CHART_SET_LEGEND@

Creates the chart's legend

Format SS_CHART_SET_LEGEND@(chartName, format chart_legend@ inLegend)

Method [this.chart_set_legend@](#)(chartName, format chart_legend@ inLegend)

Arguments chartName A chart's name.
inLegend The data for the legend.

Description Sets a chart's legend to the values within inLegend.

format chart_legend@
disabled, Boolean
uses_x_space, Boolean
uses_y_space, Boolean
text_before_sample, Boolean
arrange_by_row, Boolean
max_per_row_or_column,
title, Number
horizontal_alignment, String
left
center
right
horizontal_offset, Number
vertical_alignment, top
middle
bottom
vertical_offset,
top_margin,
bottom_margin,
row_margin,
left_margin,
right_margin,
column_margin,
proportional_margins,
title_alignment

SS_CHART_SET_MARGIN@

Sets the margins within a chart

Format SS_CHART_SET_MARGIN@(chartName, left, right, top, bottom)

Method [this.chart_set_margin@](#)(chartName, left, right, top, bottom)

Arguments

chartName	The name of the chart within the current document.
left	The chart's left margin.
right	The chart's right margin.
top	The chart's top margin.
bottom	The chart's bottom margin.

Description Sets the margin within the chart's extent. That is, these margins are within the area used by Graphics to draw the chart. The default value is 500 mils. (1000 mils - 1 inch)
Note that the title, subtitle, footer, and legend ignore these margins and are placed on the extent.

SS_CHART_SET_NULL_FORMAT@

Specifies how NULLS are displayed within a chart

Format SS_CHART_SET_NULL_FORMAT@(chartname, nullFormat)

chartname The name of chart within the current document.

formatString One of the following values:

zero

span

gap

[Method](#) this.chart_set_null_format@(chartname, nullFormat)

[Method](#) this.chart_set_null_format@(chartname, nullFormat)

Description Tells Graphics how it should chart NULL points. For more information, see [GR_CHART_GET_NULL_FORMAT@](#).

SS_CHART_SET_ORIENTATION@

Sets vertical or horizontal chart orientation

Format SS_CHART_SET_ORIENTATION@(chartName, orientation)

chartName The name of the chart within the current document.

orientation The chart's orientation. To set the orientation to vertical, set this parameter's value to "vertical". Any other value sets the orientation to horizontal.

[Method](#) this.chart_set_orientation@(chartName, orientation)

Description Sets the chart's orientation to horizontal or vertical.

SS_CHART_SET_TEMPLATE_DIR@

Sets the Spreadsheets profile for the default template directory

Format SS_CHART_SET_TEMPLATE_DIR@(pathname)

Arguments pathname A string. This string must contain an absolute pathname of the chart template files. The default is the /axhome/charts directory in your user area.

Description Sets the default chart template directory in the Spreadsheet preferences.

SS_CHART_SET_TITLE@

Sets the titles that can appear within a chart

Format SS_CHART_SET_TITLE@(chartName, format chart_titles@ inTitles)

Method [this.chart_set_title@](#)(chartName, format chart_titles@ inTitles)

chartName The name of the chart.

titles The title information being set.

Description Sets the title, subtitle, and footer as well as these element's alignment.

format chart_title@

title,

subtitle,

footer,

title_alignment,

subtitle_alignment,

footer_alignment

SS_CLEAR@

Deletes contents and style settings

Format SS_CLEAR@()

Method [this.clear@](#)

Description If no cells are selected, `SS_CLEAR@` deletes the data and style settings from the cell in which the cursor currently resides. If cells are selected, `SS_CLEAR@` deletes the data and style settings from all the selected cells. Data deleted by `SS_CLEAR@` is not saved to the clipboard and therefore is not recoverable.

See also [SS_BLANK@](#)

SS_CLEAR_PAGEBREAKS@

Removes all manual page breaks

Format `SS_CLEAR_PAGEBREAKS@(position)`

Method `this.clear_pagebreaks@(position)`

Arguments `position` A string indicating the place in a spreadsheet where a page break should be inserted. `position` can be a row, column, or combination of rows and columns. A page break will be inserted before the row or column you specify in `position`.

For example, if `position` is specified as "C," a page break is inserted before column C. If `position` is specified as "3," a page break is inserted before row 3. If `position` is specified as "C,3," a page break is inserted before column C and another page break is inserted before row 3.

Description Removes all manual page breaks that have previously been set in the current spreadsheet.

See also [SS_SET_PAGEBREAKS@](#)

SS_CLEAR_PROTECTION@

Removes protection from protected spreadsheet cells

Format `SS_CLEAR_PROTECTION@(range)`

Method `this.clear_protection@(range)`

Arguments `range` The range of cells from which to remove protection.

Description Removes protection from cells which have previously been protected by `SS_SET_PROTECTION@` or through the selection of Protected on the Style ® Protection ® Protected. After protection is removed, the cells can be edited.

See also [SS_SET_PROTECTION@](#).

SS_CLOSE_OBJECTS@

Closes all objects in the spreadsheet

Format SS_CLOSE_OBJECTS@()

Method [this.close_objects@](#)

Description Closes all open Graphics and Chart windows whose objects are embedded within the spreadsheet.

See also [SS_OPEN_OBJECT@](#)

SS_COL_NUM@

Converts spreadsheet column letter to a number

Format colNum = SS_COL_NUM@(string)

Method colNum = [this.col_num@\(string\)](#)

Arguments string The letter (or letters) that appear over a column. This string can be upper- or lowercase.

Description Converts a spreadsheet column letter name to a column number. Column numbers are 0 based. For example, SS_COL_NUM@ returns the number 28 for the column string "AC".

See Also [SS_COL_STRING@](#)

SS_COL_STRING@

Converts Spreadsheet column number to a letter

Format colString = SS_COL_STRING@(colNum)

Method colString = [this.col_string@\(colNum\)](#)

Arguments colNum The Spreadsheet column to be converted to a string.

Description Converts the numeric representation of a column to its letter (string) representation. Column numbers are 0 based. For example, `SS_COL_STRING@` returns the column letter ``C" for the column number 2.

See Also [SS_COL_NUM@](#)

SS_COORDINATE@

Converts a cell's numeric cell address to its letter (string) representation

Format `newCoord = SS_COORDINATE@(column,row[, sheet])`

Method `newCoord = this.coordinate@(column,row[, sheet])`

Arguments

column	The spreadsheet column (number) to be converted to a string (letter).
row	The spreadsheet row (number) to be converted to a string (number).
sheet	The spreadsheet sheet (number) to be converted to a string (letter).

Description Converts the numerical representation of a column and row to its string representation. Column numbers and row numbers are zero-based. For example, `SS_COORDINATE@(0,2,4)` returns the cell address E:A3.

See Also [SS_COL_NUM@](#)
[SS_COL_STRING@](#)

SS_COPY@

Copies selected data

Format `SS_COPY@([cellOrRange])`

Method `this.copy@(cellOrRange)`

Arguments `cellOrRange` A string that represents either the cell address (for example, ``A5"), the range (for example, ``A1..E15"), or the range name (for example, ``top_sector"). This argument is optional.

Description Copies selected data from the current Spreadsheets document and places it in the clipboard. `SS_COPY@` is called by the Edit ® Copy menu option.

If you call `SS_COPY@` with no arguments, information is copied to the clipboard in the following order:

1. Selected text on the edit line
2. A selected object
3. Selected cells

See also [SS CUT@](#)
[SS PASTE@](#)
[SS PASTE SPECIAL@](#)

SS_CREATE_CHART@

Creates or edits a chart

Format SS_CREATE_CHART@(format ss_chart_info_ info)

Method [this.create_chart@](#)(format ss_chart_info_ info)

Arguments info A data structure defining a chart's attributes.

Description **This macro is obsolete!** Please use [SS CHART CREATE@](#) instead.

Creates or edits a chart. The contents and structure of this chart is defined using the info, whose definition is as follows:

```
format ss_chart_info_
  task_id,          ' task id of graph if open (otherwise 0)
  name,            ' of chart
  main_type,
  dataoffset,      ' index to start of data group (after titles & axes)
  format arrayof2 ss_chart_data,
  axis_labels,     ' 1st y axis labels (2.1)
  legends,
  orient,
  path,            ' Path to TempFile/External Link
  extlink,         ' if TRUE path is to an external link
  extname,         ' name of chart in external Graphics file
  bar_overlap,
  bar_margin,
  groups_inited,  'true groups initialized for external charts only
  chart_type,     ' main chart type
  tmargin,        ' top margin
  bmargin,        ' bot margin
  lmargin,        ' left margin
  rmargin,        ' right margin
  null_format     ' span, gap, or substitute zero for null points
```

The definition of `ss_chart_` is as follows:

```
format ss_chart_  
    chart_type,          ' chart type pie, bar...  
    series,              ' all data ranges as strings !!!  
    axes,                ' list of axes  
    format arrayof2 chart_titles@ titles,  
    format arrayof2 chart_legend@ legend,  
    format arrayof2 gr_attribute@ title_attr,  
    format arrayof2 gr_attribute@ subtitle_attr,  
    format arrayof2 gr_attribute@ footer_attr,  
    format arrayof2 gr_attribute@ leg_title_attr,  
    format arrayof2 gr_attribute@ leg_box_attr,  
    format arrayof2 gr_attribute@ leg_label_attr,  
    format arrayof2 gr_attribute@ minx_attr,  
    format arrayof2 gr_attribute@ miny_attr,  
    format arrayof2 gr_attribute@ majx_attr,  
    format arrayof2 gr_attribute@ majy_attr,  
    format arrayof2 gr_attribute@ back_attr,  
    format arrayof2 chart_decorations@ decs,  
    format arrayof2 ss_chart_group@ grp,  
    format arrayof2 ss_chart_axis_info@ x_axes,  
    format arrayof2 ss_chart_axis_info@ y_axes,  
    format arrayof2 ss_chart_axis_info@ z_axes
```

The definition of `ss_chart_group@` is as follows:

```
format ss_chart_group@  
    type,  
    stack_id,  
    custom_part_name,  
    name,  
    user_str,          /* the group name as displayed to the user */  
    x_axis,  
    y_axis,  
    z_axis,  
    label_x_offset,  
    label_y_offset,  
    label_alignment,  
    label_type,  
    legend_string,  
    series,  
    main_type,  
    option,            'the % explosion value for pie charts  
    format arrayof2 gr_attribute@ attr,  
    format arrayof2 gr_attribute@ l_attr,
```

```
format arrayof2 chart_number_format@ label_format,  
format arrayof2 chart_datum@ data
```

The definition of `ss_chart_axis_info@` is as follows:

```
format ss_chart_axis_info@  
axis_str,      'name of axis  
axis_rngstr,  
format chart_axis_info@ axis_info,      'axis info struct  
format gr_attribute@ l_attr,'axis label attributes  
format gr_attribute@ t_attr,      'tick label attributes  
format gr_attribute@ axis_attr      'axis attributes
```

For the definition of:

```
gr_attribute@  
    see GR CHART GET ATTR@  
chart_datum@  
    see GR CHART SET DATUM@  
chart_decorations@  
    see GR CHART SET DECORATIONS@  
chart_legend@  
    see GR CHART SET LEGEND@  
chart_number_format@  
    see GR CHART FORMAT NUMBER@  
chart_title@  
    see GR CHART SET TITLE@
```

SS_CREATE_NAMED_RANGE@

Creates a named range

Format `SS_CREATE_NAMED_RANGE@(name, rangePos[, refCell])`

Method `this.create_named_range@(name, rangePos[, refCell])`

Arguments

name	The name to be assigned a range.
rangePos	The range's extent.
refCell	If this argument exists, you are entering the address of a reference cell for a relative range.

Description Creates a named range. That is, it associates a name with a range specification such that the range name can be used in any place that a cell or set of cell references can be used.

SS_CUT@

Cuts selected data

Format SS_CUT@(cellOrRange)

Method [this.cut@\(cellOrRange\)](#)

Arguments cellOrRange A string that represents either the cell address (for example, ``A5"), the range (for example, ``A1..E15"), or the range name (for example, ``top_sector").

Description Cuts selected data from the current spreadsheet and places the data in the clipboard. SS_CUT@ is called by the Edit ® Cut menu.

See also [SS_COPY@](#)
[SS_PASTE@](#)
[SS_PASTE_SPECIAL@](#)

SS_DATE_STRING@

Converts a numeric date value to a date string

Format date = SS_DATE_STRING@(value, format)

Method [date = this.date_string@\(value, format\)](#)

Arguments value The date value to be converted to a string. value is based on the number of days that have elapsed since January 1, 1900.

format The format desired for the date string:

- 1 Mmmm dd, yyyy
- 2 Mmm dd, yyyy
- 3 dd Mmm yy
- 4 mm/dd/yy
- 5 dd.mm.yy

6	yyyy-mm-dd
7	yy-mm-dd
8	yyyy mm dd
9	yy mm dd
10	yyyymmdd
11	yymmdd
13	dd.mm.yyyy
15	Mmmm yyyy

Description Returns a string version of the current date, in any of 13 possible formats. This macro does not use the same time table as `CURRENT_TIME@`. `SS_DATE_STRING@` is based on the number of days that have elapsed since January 1, 1900. In contrast, `CURRENT_TIME@` is based on the number of seconds that have elapsed since January 1, 1970.

`SS_DATE_STRING@` is called by the Date Style radio button group in the Style ® Numbers menu option.

See also [SS_DATE_VALUE@](#)
[DATE_FORMAT@](#)

SS_DATE_VALUE@

Converts a date string to a numeric value

Format `dateArray = SS_DATE_VALUE@(dateString)`

Method `dateArray = this.date_value@(dateString)`

Arguments `dateString` The date string being converted to a numeric value.

Description Returns a two-element array of information about a date string in a spreadsheet, as follows:

`dateArray[0]` The numeric value of the date string. value is based on the number of days that have elapsed since January 1, 1900.

`dateArray[1]` The format of the date string:

1	Mmmm dd, yyyy
2	Mmm dd, yyyy
3	dd Mmm yy
4	mm/dd/yy
5	dd.mm.yy
6	yyyy-mm-dd
7	yy-mm-dd

8 yyyy mm dd
9 yy mm dd
10 yyymmdd
11 yymmdd
13 dd.mm.yyyy
15 Mmmm yyyy

SS_DATE_VALUE@ is called by the Style radio button in the Style ® Numbers menu option.

See also [SS DATE STRING@](#)
[DATE FORMAT@](#)

SS_DB_CREATE_VIEW@

Creates a database view

Format SS_DB_CREATE_VIEW@(dbName[, colList])

Method [this.db_create_view@\(dbName\[, colList\]\)](#)

Arguments dbName The name of the database.
colList The columns that will be part of the view. Each element in the list is a string containing the column heading letter.
If you omit this parameter, all columns in the database become part of the view.

Description Creates a view into a database using the column list from the database. This macro lets you view records before you extract or delete them. It also lets you create a view that only contains columns that meet your criteria.

Use [SS SET DATABASE@](#) to create the on-sheet database.

SS_DB_DELETE_REC@

Deletes the records that satisfy a criteria

Format SS_DB_DELETE_REC@(dbName)

Method [this.db_delete_rec@\(dbName\)](#)

Arguments dbName The name of the database.

Description Deletes the records within a database that meet the criteria contained within the database's criterion range. That is, this macro deletes the records that would have been retrieved using the retrieval specification contained within the criterion range.

SS_DB_EXTRACT@

Retrieves records from a database

Format SS_DB_EXTRACT@(dbName[, extractUniqFlag[, extractClearFlag, clearRange]])

Method [this.db_extract@](#)(dbName[, extractUniqFlag[, extractClearFlag, clearRange]])

Arguments

dbName	The name of the database.
extractUniqFlag	A Boolean value which if set to TRUE indicates that only unique output records are displayed.
extractClearFlag	A Boolean value which if set to TRUE indicates that the extract range (or the portion of it set using clrRange) is cleared before new values are inserted.
clearRange	The range that is cleared if extractClearFlag is set to TRUE.

Description Extracts information from a database and places this information into the database's extract range.

If clearRange is not set and extractClearFlag is set to TRUE, no range is cleared.

SS_DELETE@

Deletes a row or column

Format SS_DELETE@(rowOrCol)

Method [this.delete@](#)(rowOrCol)

Arguments

rowOrCol	A string indicating the rows or columns to delete. rowOrCol can be: <ul style="list-style-type: none">· A single row or column.· A range of rows or columns specified in the form beginning range..end range.
----------	--

For example, to delete columns B through F, specify B..F.

Description When you delete a row or column, all the data in that row or column is deleted.

When a row is deleted, all data beneath the deleted row is shifted up one row. When a column is deleted, all data to the right of the deleted column is shifted left one column.

All cell references within formulas are automatically updated to reflect the new cell location.

See also [SS_DELETE_COLS@](#)
[SS_DELETE_ROWS@](#)
[SS_DELETE_SHEETS@](#)
[SS_INSERT@](#)

SS_DELETE_COLS@

Deletes the specified columns

Format SS_DELETE_COLS@(range[, count])

Method [this.delete_cols@](#)(range[, count])

Arguments

range	A string indicating the columns to delete. range can be: <ul style="list-style-type: none">· A single column.· A range of columns on one or more sheets specified in the form beginning range..end range.
count	The number of columns being deleted, starting from the first column in the given range. If no count is specified, all of the columns in the range are deleted.

For example, to delete columns B through F, specify B..F. To delete columns C through E on sheets A through B, specify A:C..B:E. If a sheet is not specified, columns are deleted only from the current sheet.

Description When you delete a column, all the data in that column is deleted.

When a column is deleted, all data to the right of the deleted column is shifted left one column.

All cell references within formulas are automatically updated to reflect the new cell location.

See also [SS_DELETE@](#)
[SS_DELETE_ROWS@](#)

SS_INSERT@

SS_DELETE_DB@

Deletes an on-sheet database

Format SS_DELETE_DB@(dbName)

Method [this.delete_db@](#)(dbName)

Arguments dbName The name of the database.

Description Deletes a database and all references associated with the database (such as criterion and extract ranges). The database and the data within the criterion and extract ranges is not deleted.

SS_DELETE_DOC@

Deletes the current spreadsheet

Format SS_DELETE_DOC@()

Method [this.delete_doc@](#)

Description Deletes the current spreadsheet and closes the Spreadsheets window.

SS_DELETE_KEY@

Deletes selection or removes preceding character

Format SS_DELETE_KEY@()

Method [this.delete_key@](#)

Description Deletes the current selection or removes the character preceding the cursor.

See also [SS_BACKSPACE_KEY@](#)

SS_DELETE_LINE@

Deletes to the beginning of the entry line's text

Format SS_DELETE_LINE@()

Method [this.delete_line@](#)

Description Deletes text line currently displayed in the entry area of the Spreadsheets document. The deletion begins at the current cursor position and continues to the beginning of the line.

For example, assume that the cursor is immediately after the period in the following line.
BEFORE CURSOR.AFTER CURSOR

After invoking this command, the text in the line is as follows:

AFTER CURSOR

This text will still be after the cursor.

The spreadsheet must be in edit mode for the deletion to occur. SS_DELETE_LINE@ is called by the Keys ® Delete line menu option.

See also [SS_DELETE_OBJECT@](#)

[SS_DELETE_RANGE@](#)

[SS_DELETE_TO_EOL@](#)

[SS_DELETE_WORD@](#)

[SS_DEL_VIEW@](#)

SS_DELETE_OBJECT@

Deletes an object from the spreadsheet

Format SS_DELETE_OBJECT@(objName)

Method [this.delete_object@\(objName\)](#)

Arguments objName The name of an object contained within the current spreadsheet. In most cases, this is a graphic or chart object.

Description Deletes an object from the spreadsheet. If the object was linked to the document, only the link is removed.

See also [SS_DELETE_LINE@](#)
[SS_DELETE_RANGE@](#)
[SS_DELETE_TO_EOL@](#)
[SS_DELETE_WORD@](#)
[SS_DEL_VIEW@](#)

SS_DELETE_RANGE@

Deletes information and style settings from selected cell(s)

Format SS_DELETE_RANGE@(range)

Method [this.delete_range@\(range\)](#)

Arguments range A string indicating the range of cells from which to delete information. Multiple ranges can be specified.

Description Deletes the contents and style settings for the cell(s) you specify. Only the cell contents are deleted. The deleted information is not saved in the clipboard.

See also [SS_DELETE_LINE@](#)
[SS_DELETE_OBJECT@](#)
[SS_DELETE_TO_EOL@](#)
[SS_DELETE_WORD@](#)
[SS_DEL_VIEW@](#)

SS_DELETE_ROWS@

Deletes the specified rows

Format SS_DELETE_ROWS@(range[, count])

Method [this.delete_rows@\(range\[, count \]\)](#)

Arguments range A string indicating the rows to delete. range can be:

- A single row.
- A range of rows specified in the form beginning range..end range.

For example, to delete rows 2 through 5, specify 2..5. To delete rows 1 through 4 on sheets A through B, specify A:1..B:4. If a sheet is not specified, rows are deleted only from the first sheet.

count The number of rows being deleted, starting from the first column in the given range. If no count is specified, all of the rows in the range are deleted.

Description When you delete a row, all the data in that row is deleted.

When a row is deleted, all data beneath the deleted row is shifted up one row.

All cell references within formulas are automatically updated to reflect the new cell location.

See also [SS_DELETE@](#)
[SS_DELETE_COLS@](#)
[SS_INSERT@](#)

SS_DELETE_SHEETS@

Removes sheet contents

Format SS_DELETE_SHEETS@(range, [count])

Method [this.delete_sheets@\(range \[,count\]\)](#)

Arguments range A string indicating the range of sheets from which to delete information.
count The number of sheets to delete.

Description Removes the sheets indicated by range.

If you specify a sheet address with no cell, SS_DELETE_SHEETS@ deletes the current sheet. For example, **SS_DELETE_SHEETS@a** deletes the current sheet.

If you specify a sheet address, a colon, and a cell, SS_DELETE_SHEETS@ deletes the designated sheet. For example, **SS_DELETE_SHEETS@ B:A1** deletes sheet B.

SS_DELETE_TO_EOL@

Deletes all text to the end of the line

Format SS_DELETE_TO_EOL@()

Method [this.delete_to_eol@](#)

Description Deletes all text beginning at the current cursor position and ending at the end of the line. The spreadsheet must be in edit mode in order for the deletion to occur. [SS_DELETE_TO_EOL@](#) is called by the Keys ® Delete EOL menu option.

See also [SS_DELETE_LINE@](#)
[SS_DELETE_OBJECT@](#)
[SS_DELETE_RANGE@](#)
[SS_DELETE_WORD@](#)
[SS_DEL_VIEW@](#)

[SS_DELETE_WORD@](#)

Deletes the current word

Format [SS_DELETE_WORD@\(\)](#)

Method [this.delete_word@](#)

Description Deletes all text from the current cursor position to the end of the current word in the entry area. If the cursor is in a space, the word after the space is deleted. The spreadsheet must be in edit mode for the deletion to occur. [SS_DELETE_WORD@](#) is called by the Keys ® Delete word menu option.

See also [SS_DELETE_LINE@](#)
[SS_DELETE_OBJECT@](#)
[SS_DELETE_RANGE@](#)
[SS_DELETE_TO_EOL@](#)
[SS_DEL_VIEW@](#)

[SS_DEL_VIEW@](#)

Deletes a named view

Format [SS_DEL_VIEW@\(name\)](#)

Method [this.del_view@\(name\)](#)

Arguments name The name of an existing view.

Description Deletes name from the list of named views. Deleting a view does not delete spreadsheet information. name should not be set to:

- Current since you cannot delete the view named Current from a spreadsheet.
- All since the current view is included in the All reference.

See also [SS_DELETE_LINE@](#)
[SS_DELETE_OBJECT@](#)
[SS_DELETE_RANGE@](#)
[SS_DELETE_TO_EOL@](#)
[SS_DELETE_WORD@](#)

SS_DOWN_ARROW_KEY@

Moves the selection down one cell

Format SS_DOWN_ARROW_KEY@()

Method [this.down_arrow_key@](#)

See also [SS_BACK_RETURN_KEY@](#)
[SS_LEFT_ARROW_KEY@](#)
[SS_RETURN_KEY@](#)
[SS_RIGHT_ARROW_KEY@](#)
[SS_UP_ARROW_KEY@](#)

SS_DRAG_INSET_OBJECT@

Starts drag mode for placing an inset within the spreadsheet

Format SS_DRAG_INSET_OBJECT@(format [ss_obj_info@](#) object, range[, gfx])

Method [this.drag_inset_object@\(format ss_obj_info@ object, range\[, gfx \]\)](#)

Arguments object A format contained with spreadsheet.am that defines object properties. The definition of this format is:
format [ss_obj_info@](#)

	name, type, property, hidden, locked, extlink, print, path, macro_to_run, no_border, title
range	The range that contains a chart's data. If the graphic object is not a chart, this argument is ignored. While it is ignored in this case, you should pass an argument of NULL.
gfx	A graphics handle. This argument is not used for chart objects. However, it is used for all other objects.

Description Starts drag mode for placing an inset within the spreadsheet. This macro can only be used interactively. That is, it cannot be used to insert an object at some predefined size.

If the object being drawn is a chart, you should call this macro as follows:

SS_DRAG_INSET_OBJECT@(info, range)

For other objects, use the following call:

SS_DRAG_INSET_OBJECT@(info, NULL, gfx)

SS_DRAW_CHART@

Draws a chart

Format SS_DRAW_CHART@(chartName, locArray)

Method [this.draw_chart@\(chartName, locArray\)](#)

Arguments

chartName	The name of a chart.
locArray	An array of four points, specifying the corners of the chart's drawing extent, as follows: x0 y0 x1 y1 These input positions are specified in mils. (1000 mils = 1 inch)

Description This macro is obsolete. Please use [SS_CHART_DRAW@](#) instead.
Draws (or redraws) a chart within a rectangle defined by the four points in the locArray.

SS_EDIT_SIZE_PROFILE@

Sets the default point size at which data is displayed

Format SS_EDIT_SIZE_PROFILE@(num)

[Method](#) this.edit_size_profile@(num)

Arguments num The point size that will become the default size. The default size is 14 point.

SS_EDIT_MODE@

Places the spreadsheet in edit mode

Format SS_EDIT_MODE@()

[Method](#) this.edit_mode@

SS_EMBED_INSET@

Embeds a file as an inset in a Spreadsheet

Format SS_EMBED_INSET@(pathname, docType, execMacro, filterMacro, new)

Arguments

pathname	The absolute pathname of a file to embed at the cursor location in the current Spreadsheet.
docType	An integer indicating the type of document. The valid doctypes are listed in the file recgfil_.am.
execMacro	The macro executed when you double-click the inset.
filterMacro	The filter macro converts a foreign file into Applixware format to display it in an Applixware document. You can enter an alternative file filter program you have created to convert the file instead of the Applixware-supplied filter. Leave this option blank to use the default conversion macro.

new -1 - open the document indicated by pathname with the appropriate Applixware application, using the filter macro.
 0 - embed the document indicated by pathname in the spreadsheet.

Description Embeds a file in a Spreadsheet at the cursor location. If new = -1, the target document is opened in the appropriate Applixware application so that it can be edited. If new = 0, the document is embedded with no editing session.

SS_ENTER_KEY@

Does a RETURN

Format SS_ENTER_KEY@()

Method [this.enter_key@](#)

Description Performs a RETURN, which terminates data entry into a cell. Use the **SS_SETCR VALUE@** macro to specify the cell into which the cursor is placed after the RETURN executes.

SS_ESCAPE_KEY@

Exits from edit mode

Format SS_ESCAPE_KEY@()

Method [this.escape_key@](#)

Description If you are in edit mode, exits from edit mode. If you are displaying a dialog box, this macro is the same as the Cancel button.

SS_EXIT@

Closes the Spreadsheets window

Format SS_EXIT@()

Method [this.exit@](#)

Description Closes the current Spreadsheets window. This macro does not save the contents of the current spreadsheet or display a dialog box asking if you want to save the spreadsheet.

This means that any edits made to the spreadsheet since the last time it was saved are lost.

If, however, more than one window is open into the same Spreadsheets document, this macro simply removes the window.

SS_EXTRACT_CELL@

Returns the numeric equivalent of a string cell address

Format numAddrArray = SS_EXTRACT_CELL@(address)

Method numAddrArray = this.extract_cell@(address)

Arguments address The string representing the cell whose address is to be converted to a numerical representation.

Description Returns a three-element array whose elements contain the numerical representation of a cell's address. For example, SS_EXTRACT_CELL@("B:A14") returns a three-element array as follows:

array[0]	column	0
array[1]	row	13
array[2]	sheet	1

See also [SS_EXTRACT_RANGE_INFO@](#)

SS_EXTRACT_COL_ROW@

Returns the numeric equivalent of a string cell address

Format numAddrArray = SS_EXTRACT_COL_ROW@(address)

Method numAddrArray = this.extract_col_row@(address)

Arguments address The string representing the cell whose address is to be converted to a numerical representation.

Description Returns a three-element array whose elements contain the numerical representation of a cell's address. For example, SS_EXTRACT_COL_ROW@("B:A14") returns a three-element array as follows:

array[0]	column	0
array[1]	row	13
array[2]	sheet	1

This macro accepts relative cell addresses only. It does not accept absolute cell addresses. For example, `SS_EXTRACT_COL_ROW@(A:A12)` works, while `SS_EXTRACT_COL_ROW@($A:$A12)` returns the error A Cell Coordinate is required here.

See also [SS_EXTRACT_RANGE_INFO@](#)

SS_EXTRACT_RANGE_INFO@

Extracts column and row numbers from a range string

Format `infoArray = SS_EXTRACT_RANGE_INFO@(range)`

Method `infoArray = this.extract_range_info@(range)`

Arguments `range` A range specification; for example; "a1..c5".

Description Returns a numeric array representing the starting and ending cells of the specified range.

Column letters are represented as numbers, with column A being number 0, column B being number 1, and so on. Rows are numbered from 0, with row 1 being 0, row 2 being 1, and so on.

The returned array elements are as follows:

`infoArray[0]` The column number for the top left cell in the range.

`infoArray[1]` The row number for the top left cell in the range.

`infoArray[2]` The column number for the bottom right cell in the range.

`infoArray[3]` The row number for the bottom right cell in the range.

`infoArray[4]` The sheet number for the first sheet in the range.

`infoArray[5]` The sheet number for the last sheet in the range.

See also [SS_EXTRACT_COL_ROW@](#)

SS_FILL@

Fills a range with numbers

Format `SS_FILL@ (range, startNumber, increment[, stopValue])`

Method `this.fill@ (range, startNumber, increment[, stopValue])`

Arguments	range	A string indicating the range of cells to fill.
	startNumber	The number to place in the first cell of the range.
	increment	The number to increment each time a new cell is filled. increment is initially added to startNumber to obtain the number for the second cell. The value of each subsequent cell is incremented by increment to obtain the value for the next cell in the range. If increment is a negative number, the cell values are decremented by the specified amount.
	stopValue	An optional number indicating a value not to be exceeded by the fill operation. If the stopValue is reached before range is filled, no further cells in range are filled. A fill will not extend beyond range; if all the cells in range are filled before the stopValue is reached, the stopValue is ignored.

Description Insert numbers into the cells of a range. That is, the cells are filled with numbers whose values change in the way specified here. range is filled by filling columns from top to bottom. Numbers begin with startNumber and are incremented by increment until the end of range or the stopValue are reached.

You can enter dates as startNumber and stopValue entries to build a table of dates. Date values are always incremented by days, so you must enter 1 as the increment value.

SS_FIND_GRAPHIC_OBJECT@

Moves to a Graphic object

Format SS_FIND_GRAPHIC_OBJECT@(objName)

Method [this.find_graphic_object@\(objName\)](#)

Arguments objName The name of the graphic object.

Description Moves the cell cursor to the location of the named graphic object within the Spreadsheet.

SS_FIND_REPLACE@

Performs a search and/or replace operation

Format SS_FIND_REPLACE@(cmd, srchText, repText, matchCaseFlag, wholeCellFlag, colWiseFlag, searchInRangeFlag, searchRange, valuesFlag)

Method [this.find_replace@](#)(cmd, srchText, repText, matchCaseFlag, wholeCellFlag, colWiseFlag, searchInRangeFlag, searchRange, valuesFlag)

Arguments

cmd	One of the following values: 0 find next 1 replace 2 replace all
srchText	The string to be found.
repText	If srchText is found, the text to replace it with.
matchCaseFlag	A Boolean value which if set to TRUE indicates that a match will occur if the case of all letters in srchText are the same as the text within a cell. FALSE indicates that the case of the text within the cell can differ.
wholeCellFlag	A Boolean value which if set to TRUE indicates that a match will only occur if the contents of the srchText completely matches the entire contents of the cell. FALSE indicates that other text can be within the cell.
colWiseFlag	A Boolean value which if set to TRUE indicates that searches are performed column by column. If this flag is FALSE, searches are performed row by row.
searchInRangeFlag	A Boolean value which if set to TRUE indicates that searches are to be constrained to the range indicated in searchRange.
searchRange	If searchInRangeFlag is TRUE, the search will be constrained to this range.
valuesFlag	A Boolean value which if set to TRUE indicates that searches are made on a cell's value rather than its contents; that is, it will search for the result of a formula rather than the formula's text.

Description Searches for text within the Spreadsheet and, if the text is found, replaces it.

SSF_AVEDEV@

Calculates average of the absolute deviations of data points from their means

Format SSF_AVEDEV@(number1, number2...)

Arguments number1,number2... A numeric list of values.

Description Calculates the average of the absolute deviations of data points from their means. SSF_AVEDEV@ is a measure of variability in a data set.

Text and empty cells are ignored in calculations.

If you do not provide numeric arguments, SSF_AVDEV@ returns "Not enough arguments."

SSF_BETADIST@

Evaluates the cumulative probability density function

Format BETADIST@(x, alpha, beta, A, B)

Arguments

x	The value at which the function is evaluated over the interval $A \leq x \leq B$.
alpha	A positive numeric value representing a parameter to the distribution.
beta	A positive numeric value representing a parameter to the distribution.
A	The optional lower bound to the interval of x. The value must be numeric.
B	An optional upper bound to the interval of x. The value must be numeric.

Description Evaluates the cumulative beta probability density function.

- If $x < A$, $x > B$, or $A = B$, SSF_BETADIST@ returns "Arguments out of range."
- If you do not include values for the A and B arguments, SSF_BETADIST@ uses the standard cumulative beta distribution ($A = 0$ and $B = 1$).

See also [SSF_BETAINV@](#)

SSF_BETAINV@

Evaluates the inverse of the cumulative beta probability density function

Format SSF_BETAINV@(probability, alpha, beta, A,B)

Arguments

probability	The probability for which the inverse of the beta distribution function is evaluated.
alpha	A positive numeric value representing a parameter to the distribution.
beta	A positive numeric value representing a parameter to the distribution.
A	The optional lower bound to the interval of x. The value must be numeric.
B	An optional upper bound to the interval of x. The value must be numeric.

Description SSF_BETAINV@ evaluates the inverse of the cumulative beta probability density function. For example, if probability = SSF_BETADIST@(x,...) then SSF_BETAINV@(probability,...) = x.

- If $x < A$, $x > B$, or $A = B$, SSF_BETADIST@ returns "Arguments out of range."
- If you do not include the A and B arguments, SSF_BETAINV@ uses the standard cumulative beta distribution ($A = 0$ and $B = 1$).
- SSF_BETAINV@ will calculate iteratively until the result is accurate to within $\pm 3 \cdot 10^{-7}$. If SSF_BETAINV@ does not converge after 100 iterations, it returns an error.
- If probability ≤ 0 or probability > 1 , SSF_BETAINV@ returns "Arguments out of range."

See also [SSF_BETADIST@](#).

SSF_BINMODIST@

Evaluates the individual term binomial distribution probability

Format SSF_BINMODIST@(number_s, trials, probability_s, cumulative)

Arguments

number_s	An integer representing the number of successes in the trials.
trials	An integer representing the number of independent trials.
probability_s	A numeric value representing the probability of success on each trial.
cumulative	A logical value that determines the form of the function. If cumulative is TRUE, SSF_BINMODIST@ will return the cumulative distribution function. This is the probability that there are at the most number_s successes. If

cumulative is FALSE, SSF_BINMODIST@ will return the probability density function. This is the probability that there are number_s successes.

Description SSF_BINMODIST@ evaluates the individual term binomial distribution probability. The function returns the probability that number_s or fewer successes will occur in a specified number of independent trials, each of which has a probability_s of success.

- If probability_s is <0 or >1, SSF_BINMODIST@ returns "Arguments out of range."
- If number_s is < 0 or > trials, SSF_BINMODIST@ returns "Arguments out of range."

SSF_BIVARNORMDIST@

evaluates the normal cumulative joint distribution function for the specified pairs of mean and standard deviations

Format SSF_BIVARNORMDIST@(x, x_mean, x_sdev, y, y_mean, y_sdev, corr_coef, cumulative)

Arguments

- x, y The numeric values for which you want the joint distribution.
- x_mean, y_mean The arithmetic means of the distribution.
- x_sdev, y_sdev The standard deviations of the distribution.
- corr_coef The correlation coefficient between x and y.
- cumulative A logical value that determines the form of the function. If cumulative is TRUE, SSF_BIVARNORMDIST@ returns the cumulative joint distribution function. If cumulative is FALSE, SSF_BIVARNORMDIST@ returns the joint distribution function.

Description SSF_BIVARNORMDIST@ evaluates the normal cumulative joint distribution function for the specified pairs of mean and standard deviations.

Equation

$$\left[\frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{1}{2} \frac{(x-\mu_x)^2}{\sigma_x^2(1-\rho^2)}} \right] \left[\frac{1}{\sigma_y \sqrt{2\pi}} e^{-\frac{1}{2} \frac{(y-\mu_y)^2}{\sigma_y^2(1-\rho^2)}} \right] \left[\frac{1}{\sqrt{1-\rho^2}} e^{-\frac{\rho(x-\mu_x)(y-\mu_y)}{\sigma_x \sigma_y (1-\rho^2)}} \right]$$

Description If x_sdev or y_sdev is less than or equal to zero, SSF_BIVARNORMDIST@ returns Arguments out of range.

If `corr_coef` is less than or equal to zero, `SSF_BIVARNORMDIST@` returns Arguments out of range.

The following returns the value 0.0849899.

`SSF_BIVARNORMDIST@ (3,5,1.5,5,2,3,0.25,TRUE)`

See also [SSF_NORMDIST@](#)

SSF_CHIDIST@

Evaluates the chi-squared distribution function

Format `SSF_CHIDIST@(x, degreesFreedom)`

Arguments `x` The probability for which the chi-squared distribution function is to be evaluated. `x` must be a numeric value.

`degreesFreedom`

A numeric value representing the number of degrees of freedom of the chi-squared distribution. The value is truncated to an integer if necessary.

Description `SSF_CHIDIST@` evaluates the chi-squared distribution function. It returns the one-tailed probability of the chi-squared distribution. The χ^2 distribution is associated with a χ^2 test.

- If `degrees_freedom < 1`, `SSF_CHIDIST@` returns "Arguments out of range."
- `SSF_CHIDIST@` will calculate its result iteratively until the result is accurate to within $\pm 3 \cdot 10^{-7}$. If `SSF_CHIDIST@` does not converge after 100 iterations, it will return an error value.

See also [SSF_CHIINV@](#)

SSF_CHITEST@

Returns the test for independence

Format `SSF_CHITEST@(actualRange, expectedRange)`

Arguments `actualRange` The range of data that contains the observations you want to test against anticipated values.

expectedRange

The range of data that contains the ratio of the product of row totals and column totals to the grand total.

Description SSF_CHITEST@ returns the test for independence. This is the value from the chi-squared distribution for the statistic and the appropriate degrees of freedom. The actual_range and expected_range must contain the same number of data points. Both actual_range and expected_range must contain more than one row or column.

The χ^2 test initially calculates a χ^2 statistic. It then sums the differences of actual values from expected values. The χ^2 statistic is:

Aij The actual frequency in the i-th row, j-th column

Eij The expected frequency in the i-th row, j-th column

r The number of row

c The number of columns

SSF_CHITEST@ returns the probability for a χ^2 statistic and degrees of freedom, df, where $df = (r - 1)(c - 1)$.

SSF_CHIINV@

Evaluates the inverse of the chi-based distribution function.

Format SSF_CHIINV@(probability,degreesFreedom)

Arguments probability The probability for which the inverse of the chi-squared distribution function is to be evaluated. probability must be a numeric value.

degreesFreedom

A numeric value representing the number of degrees of freedom. The value is truncated to an integer if necessary.

Description SSF_CHIINV@ evaluates the inverse of the chi-based distribution function. If the probability value is equal to SSF_CHIDIST@(x,...), then SSF_CHIINV@(probability,...) is equal to x.

- SSF_CHIINV@ will calculate iteratively until the result is accurate to within $\pm 3 \cdot 10^{-7}$. If SSF_CHIINV@ does not converge after 100 iterations, it will return an error.

- If degrees_freedom < 1, SSF_CHIINV@ returns "Arguments out of range."

See also [SSF_CHIDIST@](#)

SSF_CONFIDENCE@

Calculates the confidence interval for a population mean

Format SSF_CONFIDENCE@(alpha, standardDev, size)

Arguments

alpha	The significance level used to compute the confidence level. Confidence level equals 100(1 - alpha)%.
standardDev	The population standard deviation from the data range. It is assumed that this value is known. The value is truncated to an integer, if necessary.
size	An integer representing the sample size.

Description SSF_CONFIDENCE@ calculates the confidence interval for a population mean. The interval is a range on either side of a sample mean.

- If alpha is £ or ³ 1, SSF_CONFIDENCE@ returns "Arguments out of range."
- If standard_dev is £ 0, SSF_CONFIDENCE@ returns "Arguments out of range."
- If size < 1, SSF_CONFIDENCE@ returns "Arguments out of range."

SSF_CORREL@

Returns the correlation coefficient of two arrays

Format SSF_CORREL@(array1,array2)

Arguments

array1	A cell range containing numbers, range names, or cell references containing numbers.
array2	A cell range containing numbers, range names or cell references containing numbers.

Description Returns the correlation coefficient of the array1 and array2 cell ranges.

Empty cells and cells containing text are ignored.

- array1 and array2 must contain the same number of data points.
- If either array1 or array2 are empty, or if the standard deviation of their values equals 0, SSF_CORREL@ returns "Division by zero."

SSF_COVAR@

Calculates covariance

Format SSF_COVAR@(array1, array2)

Arguments array1 A cell range containing numbers, range names, or cell references containing numbers.
array2 A cell range containing numbers, range names, or cell references containing numbers.

Description SSF_COVAR@ calculates covariance. Covariance is the average of the products of deviations for each data point pair.

- Empty cells and cells containing text are ignored in calculations.
- array1 and array2 must contain the same number of data points.
- If either array1 or array2 is empty, or if the standard deviation of their values equals 0, SSF_COVAR@ returns "Division by zero."

SSF_CRITBINOM@

Calculates the smallest integer k for which the cumulative binomial distribution function is greater than or equal to the criterion alpha

Format SSF_CRITBINOM@(trials, probability_s, alpha)

Arguments trials The number of Bernoulli trials. The value is truncated to an integer if necessary.
probability_s A numeric value representing the probability of a success on individual trials.
alpha The criterion value. alpha must be a numeric value.

Description SSF_CRITBINOM@ calculates the smallest integer k for which the cumulative binomial distribution function is greater than or equal to the criterion alpha.

- If trials is ≤ 0 , SSF_CRITBINOM@ returns "Arguments out of range."
- If alpha is < 0 or > 1 , SSF_CRITBINOM@ returns "Arguments out of range."
- If probability_s < 0 or > 1 , SSF_CRITBINOM@ returns "Arguments out of range."

SSF_DEVSQ@

Calculates the sum of squares of deviations of data points from sample means

Format SSF_DEVSQ@(number1, number2...)

Arguments number1,number2 A list of numeric values.

Description SSF_DEVSQ@ calculates the sum of squares of deviations of data points from their sample mean.

Text and empty cells are ignored in calculations.

SSF_EXPONDIST@

Evaluates the exponential distribution function

Format SSF_EXPONDIST@(x, lambda, cumulative)

Arguments

x	The value of the function.
lambda	The parameter value.
cumulative	A logical value that determines which form of the exponential function to provide. TRUE returns the cumulative distribution function. FALSE returns the probability density function.

Description SSF_EXPONDIST@ evaluates the exponential distribution function. It can be used to model time between events.

- If x is < 0, SSF_EXPONDIST@ returns "Arguments out of range."
- If lambda <= 0, SSF_EXPONDIST@ returns "Arguments out of range."

SSF_FDIST@

Evaluates the *F* probability distribution

Format SSF_FDIST@(x, degreesFreedom1, degreesFreedom2)

Arguments x The value at which to evaluate the function. x must be a positive number.
degreesFreedom1

A numeric value representing the numerator degrees of freedom. The value is truncated to an integer if necessary.

degreesFreedom2

The denominator degrees of freedom. `degrees_freedom2` must be a numeric value and is truncated to an integer if necessary.

Description `SSF_FDIST@` evaluates the F probability distribution. This function can be used to determine whether two data sets have different degrees of diversity. If `x` is negative, `SSF_FDIST@` returns "Arguments out of range."

See also [SSF_FINV@](#)

SSF_FINV@

Calculates the inverse of the *F* cumulative distribution

Format `SSF_FINV@(probability, degreesFreedom1, degreesFreedom2)`

Arguments `probability` A numeric value representing a probability associated with the F cumulative distribution.

`degreesFreedom1`

A numeric value representing the numerator degrees of freedom. The value is truncated to an integer if necessary.

`degreesFreedom2`

A numeric value representing the denominator degrees of freedom. The value is truncated to an integer if necessary.

Description `SSF_FINV@` calculates the inverse of the F probability distribution.

- If `probability` is < 0 or > 1 , `SSF_FINV@` returns "Arguments out of range."
- If `degrees_freedom1` is < 1 , `SSF_FINV@` returns "Arguments out of range."
- If `degrees_freedom2` is < 1 , `SSF_FINV@` returns "Arguments out of range."
- `SSF_FINV@` will calculate iteratively until the result is accurate to within $\pm 3 \times 10^{-7}$. If `SSF_FINV@` does not converge after 100 iterations, it will return an error value.

SSF_FISHER@

Calculates the Fisher transformation at x

Format SSF_FISHER@(x)

Arguments x A numeric value.

Description SSF_FISHER@ calculates the Fisher transformation at x. The transformation creates a function that is normally distributed (approximately) rather than skewed. You can use this function to perform testing on the correlation coefficient.

If x ≤ -1 or ≥ 1 . SSF_FISHER@ returns "Arguments out of range."

See also [SSF_FISHERINV@](#)

SSF_FISHERINV@

Calculates the inverse of the Fisher transformation

Format SSF_FISHERINV@(y)

Arguments y The numeric value on which the inverse transformation is performed.

Description SSF_FISHERINV@ calculates the inverse of the Fisher transformation. You can use this transformation when you want to analyze correlations between ranges of data.

See also [SSF_FISHER@](#)

SSF_FORECAST@

Calculates a value for x based on a linear regression of x- and y- arrays

Format SSF_FORECAST@(x, known_y,known_x)

Arguments x The data point for which you want to predict a value.

known_y The dependent range of data.

known_x The independent range of data.

Description SSF_FORECAST@ calculates a value for x based on a linear regression of x- and y-ranges containing numeric data. SSF_FORECAST@ can be used to predict future trends in sales, inventory requirements, and so on.

- If the variance of known_x's equals 0, SSF_FORECAST@ returns "Division by zero."
- If known_y's and known_x's contain a different number of data points or are empty, SSF_FORECAST@ returns "Not enough arguments."

SSF_FTEST@

Calculates the results of an *F*-test

Format SSF_FTEST@(array1, array2)

Arguments array1 The first numeric range of data.
array2 The second numeric range of data.

Description SSF_FTEST@ calculates the results of an F-test. An F-test calculates the one-tailed probability that variances in array1 and array2 are not significantly different.

- If the number of data points in either range is less than 2, or if the variance of either range is zero, SSF_FTEST@ returns "Division by zero."

SSF_GAMMADIST@

Evaluates the gamma distribution function.

Format SSF_GAMMADIST@(x, alpha, beta, cumulative)

Arguments x The value at which you want to evaluate the distribution. x must be a numeric value greater ³ 0.
alpha A numeric value representing a parameter to the distribution.
beta A numeric value representing a parameter to the distribution.
cumulative A logical value that determines the function's form. If cumulative is TRUE, SSF_GAMMADIST@ returns the cumulative distribution function. If cumulative is FALSE, it returns the probability mass function.

Description SSF_GAMMADIST@ evaluates the gamma distribution function. You can use the gamma distribution function to study variables that may have a skewed distribution.

- For a positive integer n, when alpha = n/2, beta = 2, and cumulative = TRUE, SSF_GAMMADIST@ returns (1 - CHIDIST(x)) with n degrees of freedom.
- If alpha or beta are £ 0, SSF_GAMMADIST@ returns "Arguments out of range."
- When alpha=1, GAMMADIST returns the exponential distribution:
- $\lambda = \frac{1}{\beta}$
- If x < 0, SSF_GAMMADIST@ returns "Arguments out of range."
- If alpha £ 0 or if beta £ 0, SSF_GAMMADIST@ returns "Arguments out of range."

See also [SSF_GAMMAINV@](#)

SSF_GAMMAINV@

Calculates the inverse of the gamma cumulative distribution function

Format SSF_GAMMAINV@(probability, alpha, beta)

- Arguments**
- | | |
|-------------|---|
| probability | A numeric value representing the probability associated with the gamma distribution. |
| alpha | A numeric value representing a parameter to the distribution. |
| beta | A numeric value representing a parameter to the distribution. If beta = 1 the function returns the standard gamma distribution. |

- Description** SSF_GAMMAINV@ calculates the inverse of the gamma cumulative distribution function. If s = SSF_GAMMADIST@(x,...), then SSF_GAMMAINV@(s,...)= x
- SSF_GAMMAINV@ will calculate its result iteratively until the result is accurate to within ± 3'10-7. If SSF_GAMMAINV@ does not converge after 100 iterations, it will return an error.
 - If probability < 0 or probability > 1, SSF_GAMMAINV@ returns "Arguments out of range."
 - If alpha is £ 0 or if beta is £ 0, SSF_GAMMAINV@ returns "Arguments out of range."

See also [SSF_GAMMADIST@](#)

SSF_GAMMALN@

Calculates the natural logarithm of the gamma function

Format SSF_GAMMALN@(x)

Arguments x The numeric value for which you want to calculate SSF_GAMMALN@.

Description SSF_GAMMALN@ calculates the natural logarithm of the gamma function, G(x).
If x is £ 0, SSF_GAMMALN@ returns "Arguments out of range."

SSF_GEOMEAN@

Returns the geometric mean of a range of data

Format SSF_GEOMEAN@(number1, number2...)

Arguments number1,number2... A list of numeric values.

Description SSF_GEOMEAN@ returns the geometric mean of a data range.
The equation for the geometric mean is:

$$GM_y = \sqrt[n]{y_1 y_2 y_3 \dots y}$$

- Values must not be £ 0.
- Empty cells and text are ignored in calculations.

SSF_HARMEAN@

Calculates the harmonic mean of a data set

Format SSF_HARMEAN@(number1, number2,...)

Arguments number1,number2... A list of numeric values.

Description SSF_HARMEAN@ calculates the harmonic mean of a data set. Harmonic mean is the reciprocal of the arithmetic mean of reciprocals. It is always less than the geometric mean, which is less than the arithmetic mean.

The equation for the harmonic mean is:

$$\frac{1}{H_y} = \frac{1}{n} \sum \frac{1}{Y_i}$$

- Values must not be £ 0.
- Text and empty cells are ignored in calculations.

SSF_HYPGEOMDIST@

Evaluates the hypergeometric distribution

Format SSF_HYPGEOMDIST@(sampleS, numberSample, populationS, numberPopulation)

Arguments sampleS An integer representing the number of successes in the sample.

numberSample

 An integer representing the size of the sample.

populationS An integer representing the number of successes in the population.

numberPopulation

 An integer representing the population size.

Description SSF_HYPGEOMDIST@ evaluates the hypergeometric distribution. SSF_HYPGEOMDIST@ returns the probability of a given number of sample successes, given the sample size, population successes, and population size. The equation for hypergeometric distribution is:

$$P(X = x) = h(x; n, M, N) = \frac{\binom{M}{x} \binom{N-M}{n-x}}{\binom{N}{n}}$$

- If sample_s is £ 0 or greater than the lesser of number_sample or population_s, SSF_HYPGEOMDIST@ returns "Arguments out of range."
- If sample_s is less than the larger of 0 or (number_sample - number_population + population_s), SSF_HYPGEOMDIST@ returns "Arguments out of range."
- If number_sample is £ 0 or number_sample > number_population, SSF_HYPGEOMDIST@ returns "Arguments out of range."

- If population_s is £ 0 or population_s > number_population, SSF_HYPGEOMDIST@ returns "Arguments out of range."
- If number_population is £ 0, SSF_HYPGEOMDIST@ returns "Arguments out of range."

SSF_INTERCEPT@

Returns the intercept of the linear regression line through data points in known_y's and known_x's

Format SSF_INTERCEPT@(knownY, knownX)

Arguments knownY A numeric value representing a dependent set of observations.
 knownX A numeric value representing an independent set of observations.

Description SSF_INTERCEPT@ returns the intercept of the linear regression line through data points in known_y's and known_x's. The intercept is the point at which the values in the two data groups intersect the y-axis.

The equation for the intercept of the regression line is:

$$a = \bar{Y} - b\bar{X}$$

The slope is calculated as:

$$b = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2}$$

- knownY and knownX must contain the same number of data points.
- Empty cells and text are ignored in calculations.

SSF_KURT@

Calculates the kurtosis of a data set

Format SSF_KURT@(number1, number2...)

Arguments number1,number2... A list of numeric values separated by commas.

Description SSF_KURT@ calculates the kurtosis of a set of data. Kurtosis characterizes the relative peakedness or flatness of a distribution compared to the normal distribution. Positive kurtosis indicates a peaked distribution. Negative kurtosis indicates a flat distribution.

The equation for Kurtosis is:

$$\left\{ \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum \left(\frac{x_i - \bar{x}}{\sigma} \right)^4 \right\} - \frac{3(n-1)^2}{(n-2)(n-3)}$$

σ is the sample standard deviation.

- Empty cells and text are ignored in calculations.
- If there are less than 4 data points, SSF_KURT@ returns "Division by zero."
- If the standard deviation of the sample equals 0, SSF_KURT@ returns "Division by zero."

SSF_LARGE@

Returns the k-th largest value in a data set

Format SSF_LARGE@(array, k)

Arguments array A data range for which you want to retrieve the k-th largest value.
k The position from the largest value in the range of data.

Description SSF_LARGE@ returns the k-th largest value in a set of data. SSF_LARGE@ can be used to select a value based on its relative standing. For example, you can use SSF_LARGE@ to return the highest score in a set of test scores.

- If array is empty, SSF_LARGE@ returns "Not enough arguments."
- If $k \leq 0$ or if k is greater than the number of data points, SSF_LARGE@ returns "Arguments out of range." For example, if you have 10 data points, and $k = 11$, SSF_LARGE@ "Not enough arguments."

SSF_LOGINV@

Calculates the inverse of the lognormal cumulative distribution function of x

Format SSF_LOGINV@(probability, mean, standardDev)

Arguments

probability	A numeric value representing a probability associated with the lognormal distribution.
mean	The mean of ln(x). mean must be a numeric value.
standardDev	The standard deviation of ln(x).

Description SSF_LOGINV@ calculates the inverse of the lognormal cumulative distribution function of x, where ln(x) is normally distributed with parameters mean and standardDev.

The SSF_LOGINV@ equations is:

$$\text{LOGINV}(p, \mu, \sigma) = e^{(\mu + \sigma \times (\text{NORMSINV}(p)))}$$

- If probability is < 0 or probability is > 1, SSF_LOGINV@ returns "Arguments out of range."
- If standard_dev is £ 0, SSF_LOGINV@ returns "Arguments out of range."

See also [SSF_LOGNORMDIST@](#)

SSF_LOGNORMDIST@

Calculates the lognormal cumulative distribution function of x

Format SSF_LOGNORMDIST@(x, mean, standardDev)

Arguments

x	The numeric value at which the function is evaluated.
mean	A numeric value representing the mean of ln(x).
standardDev	A numeric value representing the standard deviation of ln(x).

Description SSF_LOGNORMDIST@ calculates the lognormal cumulative distribution function of x, where ln(x) is normally distributed with the parameters mean and standardDev.

The lognormal cumulative distribution equation is:

$$\text{NORMDIST}\left(\frac{\ln(x) - \mu}{\sigma}\right)$$

- If x is £ 0 or standard_dev is £ 0, SSF_LOGNORMDIST@ returns "Arguments out of range."

See also [SSF_LOGINV@](#)

SSF_MODE@

Returns the most frequently occurring value in a data range

Format SSF_MODE@(number1, number2,...)

Arguments number1, number2... A list of numeric values.

Description SSF_MODE@ returns the most frequently occurring value in a data range.

- Text and empty cells are ignored in calculations.
- If the data contains no duplicate values, SSF_MODE@ returns "#N/A".

SSF_NEGBINOMDIST@

Calculates the negative binomial distribution

Format SSF_NEGBINOMDIST@(number_f, number_s, probability_s)

Arguments number_f An integer representing the number of failures.
 number_s An integer value representing the threshold number of successes.
 probability_s The probability of a success.

Description SSF_NEGBINOMDIST@ calculates the negative binomial distribution. It returns the probability that there will be a certain number of failures before the number_s-th success, when the constant probability of success is probability_s.

The equation for negative binomial distribution is:

$$nb(x;r,p) = \binom{x+r-1}{r-1} p^r (1-p)^x$$

x is number_f, r is number_s and p is probability_s.

- If (number_f + number_s - 1) £ 0, SSF_NEGBINOMDIST@ returns "Arguments out of range."

- If probability_s is < 0 or > 1, SSF_NEGBINOMDIST@ returns "Arguments out of range."

SSF_NORMDIST@

Evaluates the normal cumulative distribution function for mean and standard deviation

Format SSF_NORMDIST@(x, mean, standard_dev)

Arguments

x	The numeric value for which you want the distribution.
mean	The arithmetic mean of the distribution.
standard_dev	The standard deviation of the distribution.

Description SSF_NORMDIST@ evaluates the normal cumulative distribution function for the specified mean and standard deviation.

The equation for the standard normal density function is:

$$f(x; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

- If mean = 0 and standard_dev = 1, SSF_NORMDIST@ returns the standard normal distribution, SSF_NORMSDIST@.
- If standard_dev £ 0, SSF_NORMDIST@ returns "Arguments out of range."

SSF_NORMINV@

Calculates the inverse of the normal cumulative distribution for standard deviation and mean

Format SSF_NORMINV@(probability, mean, standardDev)

Arguments

probability	A numeric value representing a probability to the distribution.
mean	The arithmetic mean of the distribution.
standardDev	The standard deviation of the distribution.

Description SSF_NORMINV@ calculates the inverse of the normal cumulative distribution for a specified standard deviation and mean.

- If probability is < 0 or > 1, SSF_NORMINV@ returns the "Arguments out of range."

- If standard_dev is £ 0, SSF_NORMINV@ returns "Arguments out of range."
- SSF_NORMINV@ uses the standard normal distribution if mean = 0 and standard_dev = 1
- SSF_NORMINV@ will calculate its result iteratively until the result is accurate to within ± 3´10-7. If SSF_NORMINV@ does not converge after 100 iterations, it will return an error.

See also [SSF_NORMSINV@](#), [SSF_NORMDIST@](#), [SSF_NORMSDIST@](#)

SSF_NORMSDIST@

Calculates the standard normal cumulative distribution function

Format SSF_NORMSDIST@(x)

Arguments x The numeric value for which you want the normal distribution.

Description SSF_NORMSDIST@ calculates the standard normal cumulative distribution function. The standard normal cumulative distribution has a mean of 0 and a standard deviation of 1.

The equation for the normal density function is:

$$f(z;0,1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$$

See also [SSF_NORMSINV@](#), [SSF_NORMDIST@](#), [SSF_NORMINV@](#)

SSF_NORMSINV@

Calculates the inverse of the standard normal cumulative distribution

Format SSF_NORMSINV@(probability)

Arguments probability A probability corresponding to the normal distribution. probability must be a numeric value.

Description SSF_NORMSINV@ calculates the inverse of the standard normal cumulative distribution. The distribution has mean of zero and a standard deviation of one.

- SSF_NORMSINV@ will calculate its result iteratively until the result is accurate to within ± 3´10-7. If SSF_NORMSINV@ does not converge after 100 iterations, it will return an error.

- If probability is < 0 or > 1, SSF_NORMSINV@ returns "Arguments out of range."

See also [SSF_NORMSDIST@](#), [SSF_NORMDIST@](#), [SSF_NORMINV@](#)

SSF_PEARSON@

Calculates the Pearson product moment correlation coefficient

Format SSF_PEARSON@(array1, array2)

Arguments array1 A set of independent numeric values.
array2 A set of dependent numeric values.

Description SSF_PEARSON@ calculates the Pearson product moment correlation coefficient, r, an index that ranges from -1.0 to 1.0 inclusive, and reflects the extent of a linear relationship between two data sets.

The r value of the regression line is:

$$r = \frac{n(\sum XY) - (\sum X)(\sum Y)}{\sqrt{\left[n\sum X^2 - (\sum X)^2 \right] \left[n\sum Y^2 - (\sum Y)^2 \right]}}$$

- array1 and array2 must have the same number of data points.
- Text and empty cells are ignored in calculations.

SSF_PERCENTILE@

Returns the value from a range at the k-th percentile

Format SSF_PERCENTILE@(array, k)

Arguments array The data range that defines relative standing. array must contain numeric values.
k The percentile value.

Description SSF_PERCENTILE@ returns the value from array at the k-th percentile.

If k is < 0 or > 1, SSF_PERCENTILE@ returns "Arguments out of range."

If k is not a multiple of 1/(n-1), SSF_PERCENTILE@ interpolates to determine the value at the kth percentile.

SSF_PERCENTRANK@

Returns the percentage range of x among the values in a data range

Arguments SSF_PERCENTRANK@(array, x, significance)

Arguments array The data range of numeric values that defines relative standing.
x The value for which you want to determine the rank.
significance An optional numeric value which identifies the precision of the percentage value returned by SSF_PERCENTRANK@. If omitted, results are rounded to three digits.

Description SSF_PERCENTRANK@ returns the percentage rank of x among the values in a data range.

- If x does not match one of the values in array, SSF_PERCENTRANK@ interpolates to return the correct percentage rank.
- If significance is < 1, SSF_PERCENTRANK@ returns "Arguments out of range."

SSF_PERMUT@

Returns the number of permutations of groups of specified objects that can be selected from a number of objects

Format SSF_PERMUT@(number, number_chosen)

Arguments number An integer that describes the number of objects. number must be greater than number_chosen.
number_chosen An integer that describes the number of objects in each permutation.

Description SSF_PERMUT@ returns the number of permutations of groups of number_chosen objects that can be selected from number.

The equation for SSF_PERMUT@ is:

$$P_{k,n} = \frac{n!}{(n-k)!}$$

- If number is £ 0, SSF_PERMUT@ returns "Arguments out of range."
- If number_chosen is < 0, SSF_PERMUT@ returns "Arguments out of range."
- If number < number_chosen, SSF_PERMUT@ returns "Arguments out of range."

SSF_POISSON@

Evaluates the Poisson probability distribution

Format SSF_POISSON@(x, mean, cumulative)

Arguments

x	A numeric value representing the number of	events.
mean	The expected numeric value.	
cumulative	A logical value that determines the form of the value returned. TRUE returns the cumulative Poisson probability that the number of random events will be between 0 and x inclusive. FALSE returns the Poisson probability function that the number of events will be equal to x.	

Description SSF_POISSON@ evaluates the Poisson probability distribution.

- If x is £ 0, SSF_POISSON@ returns "Arguments out of range."
- If mean is £ 0, SSF_POISSON@ returns "Arguments out of range."

SSF_PROB@

Returns the probability that values in x_range are between two specified values

Format SSF_PROB@(x_range, prob_range, lower_limit, upper_limit)

Arguments

x_range	A range of numeric values of x with which there are associated probabilities	
prob_range	A range containing a set of probabilities associated	with values in the x_range.
lower_limit	The lower end on the value for which you want a probability.	
upper_limit	The optional upper end on the value for which you want a probability.	

Description SSF_PROB@ returns the probability that values in x_range are between the specified lower_limit and upper_limit.

- If a value in prob_range is ≤ 0 , SSF_PROB@ returns "Arguments out of range."
- If a value in prob_range is > 1 , SSF_PROB@ returns "Arguments out of range."
- If the sum of values in prob_range is not equal to 1, SSF_PROB@ returns "Arguments out of range."
- If you do not specify an upper_limit value, SSF_PROB@ returns the probability of being equal to the specified lower_limit value.
- If x_range and prob_range contain a different number of values, SSF_PROB@ returns "Not enough arguments."

SSF_QUARTILE@

Returns a quartile from a range of numeric data

Format SSF_QUARTILE@(array, quart)

Arguments

array	A range of numeric values.
quart	A numeric value (0 through 4) which indicates which value to return. 0 returns the minimum value (or MIN value). 1 returns the first quartile (25th percentile). 2 returns the second quartile (50th percentile or MEDIAN value). 3 returns the third quartile (75th percentile). 4 returns the maximum value (or MAX value).

Description SSF_QUARTILE@ returns a quartile from a range of numeric data.

- If quart is < 0 or > 4 , SSF_QUARTILE@ returns "Arguments out of range."

SSF_RANK@

Returns the rank of a number in a range of numbers

Format SSF_RANK@(number, ref, order)

Arguments

number	The number whose rank you want to determine.
ref	A list of numbers. Non-numeric values are ignored.
order	A number specifying whether to rank values in ascending or descending order. 0 (or no value) will rank in descending order. A non-zero value will rank in ascending order.

Description SSF_RANK@ returns the rank of a number in a range of numbers. A number's rank is its size relative to the size of other values in the range. While duplicate numbers share the same rank, the ranking of subsequent higher numbers in the list is adjusted upward based on the number of duplicate values. For example in a range containing the numbers 1, 2, 4, 4, 7 and 8, 4 has a rank of three. The next highest values, 7 and 8, have a ranking of 5 and 6, respectively.

SSF_RSQ@

Returns the r^2 value of the linear regression line through points of data in known_y's and known_x's

Format SSF_RSQ@(knownY, knownX)

Arguments knownY A range of numeric values.
 knownX A range of numeric values.

Description SSF_RSQ@ returns the r^2 value of the linear regression line through points of data in known_y's and know_x's. This value is the square of the Pearson product moment correlation coefficient.

The SSF_RSQ@ equation is:

$$r = \frac{n(\sum XY) - (\sum X)(\sum Y)}{\sqrt{\left[n\sum X^2 - (\sum X)^2 \right] \left[n\sum Y^2 - (\sum Y)^2 \right]}}$$

- Text and empty cells are ignored in calculations.
- If either argument is empty or contains a different number of data points, SSF_RSQ@ returns "#NA"

SSF_SKEW@

Calculates the skewness of a distribution

Format SSF_SKEW@(number1, number2...)

Arguments number1,number2... A list of numeric values.
A list of numeric values for which you want to calculate skewness.

Description SSF_SKEW@ calculates the skewness of a distribution. Text and empty cells are ignored in calculations.

The equation for skewness is:

$$\frac{n}{(n-1)(n-2)} \sum \left(\frac{x_i - \bar{x}}{\sigma} \right)^3$$

- If there are less than three points of data or the sample standard deviation is 0, SSF_SKEW@ returns "Division by zero."

SSF_SLOPE@

Returns the slope of the linear regression line through data points

Format SSF_SLOPE@(knownY, knownX)

Arguments knownY A range of dependent data points.
knownX A range of independent data points.

Description SSF_SLOPE@ returns the slope of the linear regression line through data points in knownY and knownX.

The equation for the slope of the regression line is:

$$b = \frac{n \sum xy - \left(\sum x \right) \left(\sum y \right)}{n \sum x^2 - \left(\sum x \right)^2}$$

- If known_y and known_x contain a different number of values, SSF_SLOPE@ returns "#NA"
- Text and empty cells are ignored in calculations.

SSF_SMALL@

Returns the k-th smallest value in a data set

Format SSF_SMALL@(array, k)

Arguments array A range of numeric data.
k The position (from the smallest) in the data range.

Description SSF_SMALL@ returns the k-th smallest value in a set of data.

- If k \leq 0, SSF_SMALL@ returns "Arguments out of range."
- If k exceeds the number of data points, SSF_SMALL@ returns "Not enough arguments."

SSF_STANDARDIZE@

Calculates a normalized value from a distribution

Format SSF_STANDARDIZE@(x, mean, standard_dev)

Arguments x The value you want to normalize.
mean The arithmetic mean of the distribution.
standard_dev The standard deviation of the distribution. standard_dev must > 0.

Description SSF_STANDARDIZE@ calculates a normalized value from a distribution characterized by mean and standard_dev.

The equation for the normalized value is:

$$Z = \frac{X - \mu}{\sigma}$$

- If standard_dev is \leq 0, SSF_STANDARDIZE@ returns "Arguments out of range."

SSF_STEYX@

Returns the standard error of the regression

Format SSF_STEYX@(knownY, knownX)

Arguments knownY A range of dependent numeric data points.
 knownX A range of independent numeric data points.

Description SSF_STEYX@ returns the standard error of the regression. The standard error is a measure of the amount of error in the prediction of y for an individual x.

The equation for standard error of the predicted y is:

$$S_{y.x} = \sqrt{\left[\frac{1}{n(n-2)} \right] \left[n \sum y^2 - (\sum y)^2 - \frac{\left[n \sum xy - (\sum x)(\sum y) \right]^2}{n \sum x^2 - (\sum x)^2} \right]}$$

- Text and empty cells are ignored in calculations.
- known_x's and known_y's must have the same number of data points.

SSF_TDIST@

Returns the Student's t-distribution

Format SSF_TDIST@(x, degreesFreedom, tails)

Arguments x The numeric value at which to test the distribution.
 degreesFreedom An integer specifying the degrees of freedom.
 tails A value that determines the number of distribution tails to return. 1 returns a one-tailed distribution. 2 returns a two-tailed distribution.

Description SF_TDIST@ returns the Student's t-distribution. The t-distribution is used in hypothesis testing of small sample data sets.

The degreesFreedom must be greater than 1.

See also [SSF_TINV@](#)

SSF_TINV@

Returns the inverse of the Student's t-distribution for the specified degrees of freedom

Format SSF_TINV@(probability, degreesFreedom)

Arguments probability A value representing the probability connected with the two-tailed Student's t-distribution.

degreesFreedom
An integer representing the number of degrees of freedom to characterize the distribution.

Description SSF_TINV@ returns the inverse of the Student's t-distribution for the specified degrees of freedom.

- SSF_TINV@ will calculate its result iteratively until the result is accurate to within $\pm 3 \cdot 10^{-7}$. If SSF_TINV@ does not converge after 100 iterations, it will return an error.
- If probability is < 0 or > 1 , SSF_TINV@ returns "Arguments out of range."
- If degreesFreedom ≤ 1 , SSF_TINV@ returns "Arguments out of range."

See also [SSF_TDIST@](#)

SSF_TRIMMEAN@

Returns the mean calculated by excluding a percentage of data points from the top and bottom of a data set

Format SSF_TRIMMEAN@(array, percent)

Arguments array A data range to trim.

percent The fractional number of data points you want to exclude.

Description SSF_TRIMMEAN@ returns the mean calculated by excluding a percentage of data points from the top and bottom of a data set. SSF_TRIMMEAN@ rounds the number of data points excluded to the nearest multiple of 2. SSF_TRIMMEAN@ excludes a single value from the top and two values from the bottom.

- If percent < 0 or percent > 1 , SSF_TRIMMEAN@ returns "Arguments out of range."

SSF_TTEST@

Returns the probability associated with a Student's t-Test

Format SSF_TTEST@(array1, array2,tails, type)

Arguments

array1	The numeric value at which to test the distribution.
array2	An integer specifying the degrees of freedom.
tails	A numeric value that determines the number of distribution tails to return. 1 returns a one-tailed distribution. 2 returns a two-tailed distribution.
type	The type of t-Test to perform. 1 performs a Paired test. 2 performs a two-sample equal variance (homoscedastic). 3 performs a two-sample unequal variance (heteroscedastic).

Description SSF_TTEST@ returns the probability associated with a Student's t-Test.

- If array1 and array2 have a different number of data points, and type = 1, SSF_TTEST@ returns "Not enough arguments."
- If tails is any value other than 1 or 2, SSF_TTEST@ returns "Arguments out of range."

SSF_WEIBULL@

Evaluates the Weibull distribution

Format SSF_WEIBULL@(x, alpha, beta, cumulative)

Description SSF_WEIBULL@ evaluates the Weibull distribution.

Arguments

x	A numeric value at which to evaluate the function.
alpha	A numeric value representing a parameter to the distribution.
beta	A numeric value representing a parameter to the distribution.
cumulative	A logical value that determines the function's form. TRUE returns the cumulative distribution function. FALSE returns the probability density function.

The equation for the Weibull cumulative distribution function is:

$$f(x;\alpha,\beta) = 1 - e^{-(x/\beta)^\alpha}$$

The equation for the Weibull probability density function is:

$$f(x;\alpha,\beta) = \frac{\alpha}{\beta^\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha}$$

- If alpha is £ 0 or beta is £ 0, SSF_WEIBULL@ returns "Arguments out of range."
- If x is < 0, SSF_WEIBULL@ returns the "Arguments out of range."

SSF_ZTEST@

Calculates the two-tailed P-value of a z-test

Format SSF_ZTEST@(array, x, sigma)

Arguments

array	The data range to test.
x	The value to test
sigma	The population standard deviation. If you do not include this argument, SSF_ZTEST@ uses the sample standard deviation.

Description SSF_ZTEST@ calculates the two-tailed P-value of a z-test. This test calculates a score for x with respect to the data set, array, and returns the two-tailed probability for the normal distribution.

The z-test equation is:

$$1 - \text{NORMDIST}\left(\frac{\mu - x}{\sigma \div \sqrt{n}}\right)$$

SSF_ACCRINT@

Calculates average of the absolute deviations of data points from their means

Format SSF_ACCRINT@(issue, first_interest, settlement, rate, *par*, frequency, *basis*)

Arguments

issue	The security's issue date, expressed as a serial date number.
first_interest	The security's first interest date, expressed as a serial date number.

par The security's par value. If you omit this argument, SSF_ACCRINT@ uses \$1000 as the par value.

Description Calculates the accrued interest for a security that pays periodic interest. SSF_ACCRINT@ is calculated using the following formula:

$$SSF_ACCRINT@ = par \times \frac{rate}{frequency} \times \sum_{i=1}^{NC} \frac{A_i}{NL_i}$$

where A_i is the number of accrued days for the i th quasi-coupon period within the odd period. NC is the number of quasi-coupon periods that fit in the odd period (if this number contains a fraction, round it up to the next whole number); NL_i is the normal length in days of the i th quasi-coupon period within the odd period.

SSF_ACCRINTM@

Calculates the accrued interest for a security that pays interest at maturity

Format SSF_ACCRINTM@(issue,maturity, rate, par, basis)

Arguments

issue	The security's issue date, expressed as a serial date number.
maturity	The security's maturity date, expressed as a serial date number.
rate	See Security Function Arguments .
par	The security's par value. If you omit this argument, SSF_ACCRINTM@ uses \$1000 as the par value.
basis	See Security Function Arguments .

Description

SSF_ACCRINTM@ calculates the accrued interest for a security that pays interest at maturity.

SSF_ACCRINTM@ is calculated using the following formula:

$$SSF_ACCRINTM@ = par \times rate \times \frac{A}{D}$$

where A is the number of accrued days counted according to a monthly basis. For interest at maturity items, A is the number of days from the issue date to the maturity date. D is the Annual Year Basis.

See also [SSF_ACCRINT@](#).

SSF_COUPDAYBS@

Calculates the number of days from coupon start to settlement date

Format SSF_COUPDAYBS@(settlement, maturity, frequency, basis)

Description SSF_COUPDAYBS@ calculates the number of days from the beginning of the coupon period to the settlement date.

Arguments

settlement	See Security Function Arguments .
maturity	See Security Function Arguments .
frequency	See Security Function Arguments .
basis	See Security Function Arguments .

All arguments must be numeric, and are truncated to integers.

See also [SSF_COUPDAYS@](#), [SSF_COUPDAYSNC@](#), [SSF_COUPNCD@](#), [SSF_COUPNUM@](#), and [SSF_COUPPCD@](#).

SSF_COUPDAYS@

Calculates the number of days in the coupon period that contains the settlement

Format SSF_COUPDAYS@(settlement, maturity, frequency, basis)

Arguments

settlement	See Security Function Arguments .
maturity	See Security Function Arguments .
frequency	See Security Function Arguments .
basis	See Security Function Arguments .

Description SSF_COUPDAYS@ calculates the number of days in the coupon period that contains the settlement date.

See also [SSF_COUPDAYBS@](#), [SSF_COUPDAYSNC@](#), [SSF_COUPNCD@](#), [SSF_COUPNUM@](#), and [SSF_COUPPCD@](#).

SSF_COUPDAYSNC@

Calculates the number days from settlement date to next coupon date

Format SSF_COUPDAYSNC@(settlement, maturity, frequency, basis)

Arguments

settlement	See Security Function Arguments .
maturity	See Security Function Arguments .
frequency	See Security Function Arguments .
basis	See Security Function Arguments .

Description SSF_COUPDAYSNC@ calculates the number of days from the settlement date to the next coupon date.

See also [SSF_COUPDAYS@](#), [SSF_COUPDAYSNC@](#), [SSF_COUPNCD@](#), [SSF_COUPNUM@](#), and [SSF_COUPPCD@](#).

SSF_COUPNCD@

Calculates the next coupon date after the settlement

Format SSF_COUPNCD@(settlement, maturity, frequency, basis)

Arguments

settlement	See Security Function Arguments .
maturity	See Security Function Arguments .
frequency	See Security Function Arguments .
basis	See Security Function Arguments .

Description SSF_COUPNCD@ calculates the next coupon date after the settlement date.

See also [SSF_COUPDAYS@](#), [SSF_COUPDAYBS@](#), [SSF_COUPDAYSNC@](#), [SSF_COUPNUM@](#), and [SSF_COUPPCD@](#).

SSF_COUPNUM@

Calculates the number of coupons payable between settlement date and maturity date

Format SSF_COUPNUM@(settlement, maturity, frequency, basis)

Arguments settlement See [Security Function Arguments](#).
 maturity See [Security Function Arguments](#).
 frequency See [Security Function Arguments](#).
 basis See [Security Function Arguments](#).

Description SSF_COUPNUM@ calculates the number of coupons payable between the settlement date and the maturity date and rounds the result to the nearest whole coupon.

See also [SSF COUPDAYS@](#), [SSF COUPDAYBS@](#), [SSF COUPDAYSNC@](#), [SSF COUPNCD@](#), and [SSF COUPPCD@](#).

SSF_COUPPCD@

Calculates the coupon date previous to the settlement date

Format SSF_COUPPCD@(settlement, maturity, frequency, basis)

Arguments settlement See [Security Function Arguments](#).
 maturity See [Security Function Arguments](#).
 frequency See [Security Function Arguments](#).
 basis See [Security Function Arguments](#).

Description SSF_COUPPCD@ calculates the coupon date previous to the settlement date.

See also [SSF COUPDAYS@](#), [SSF COUPDAYBS@](#), [SSF COUPDAYSNC@](#), [SSF COUPNCD@](#), and [SSF COUPNUM@](#).

SSF_CUMIPMT@

Calculates the cumulative interest paid on a loan

Format SSF_CUMIPMT@(rate, nper, pv, start_period, end_period, type)

Arguments rate The interest rate per period.
 nper Total number of payment periods.
 pv The present value.
 start_period The first period in the calculation (periods are numbered starting at 1).
 end_period The last period in the calculation, and type is when the payments are due.

type If type is 1, payments are due at the beginning of the period; if type is 0, payments are due at the end of the period. The default for type is 0.

Description SSF_CUMIPMT@ calculates the cumulative interest paid on a loan between start_period and end_period.
Be sure to use the same time units for rate and nper. If you make monthly payments and use an annual rate, divide the annual rate by 12 for rate, and multiply the term by 12 for nper.

SSF_CUMPRINC@

Calculates the cumulative principal paid on a loan

Format SSF_CUMPRINC@(rate, nper, pv, start_period, end_period, type)

Arguments

rate	The interest rate per period.
nper	number of periods
pv	Present value.
start_period	The first period.
end_period	The last period in the calculation
type	When payments are due. If type is 1, payments are due at the beginning of the period; if type is 0, payments are due at the end of the period. The default for type is 0.

Description SSF_CUMPRINC@ calculates the cumulative principal paid on a load between start_period and end_period.
Be sure to use the same time units for rate and nper. If you make monthly payments and use an annual rate, divide the annual rate by 12 for rate, and multiply the term by 12 for nper.

See also [SSF_CUMIPMT@](#).

SSF_DB@

Calculates the depreciation of an asset using the fixed-declining balance method

Format SSF_DB@(cost, salvage, life, period, month)

Arguments

cost	The initial cost of the asset
------	-------------------------------

salvage	The asset's value at the end of its life.
life	the number of periods it will take to depreciate to the salvage value
period	The time period used to determine the depreciation allowance.
month	The number of months in the first year. If month is omitted, the default is 12.

Description SSF_DB@ calculates the depreciation of an asset for a specified period using the fixed-declining balance method.

The fixed-declining balance method calculates depreciation at a fixed rate. SSF_DB@ calculates depreciation for a period using the following formulas:

$$(cost - total\ depreciation\ from\ prior\ periods) \times rate$$

where

$$rate = 1 - \left((salvage / cost)^{(1/life)} \right)$$

rounded to three decimal places.

The first and last periods are special cases. For the first period, SSF_DB@ uses the formula:

$$(cost \times rate \times month) / 12$$

For the last period, SSF_DB@ uses the formula:

$$((cost - total\ depreciation\ from\ prior\ periods) \times rate \times (12 - month)) / 12$$

SSF_DISC@

Calculates the discount rate for a security

Format SSF_DISC@(settlement, maturity, pr, redemption, basis)

Arguments	settlement	See Security Function Arguments .
	maturity	See Security Function Arguments .
	pr	See Security Function Arguments .
	redemption	See Security Function Arguments .
	basis	See Security Function Arguments .

Description SSF_DISC@ calculates the discount rate for a security.

SSF_DISC@ is calculated using the following formula:

$$SSF_DISC@ = \frac{\text{redemption} - \text{par}}{\text{par}} \times \frac{B}{DSM}$$

where B is the number of days in a year, depending on the year basis, and DSM is the number of days between settlement and maturity.

See also [SSF_PRICEDISC@](#) and [SSF_YELDDISC@](#)

SSF_DOLLARDE@

Converts a dollar expression from a fraction to a decimal

Format SSF_DOLLARDE@(f_dollar, fraction)

Arguments f_dollar A number expressed as a fraction.
fraction An integer used as the fraction denominator.

Description SSF_DOLLARDE@ converts a dollar price expressed as a fraction into a dollar price expressed as a decimal number. Use SSF_DOLLARDE@ to convert fractional dollars, such as prices for securities, into decimal dollars.

See also [SSF_DOLLARFR@](#)

SSF_DOLLARFR@

Converts a dollar expression from a decimal to a fraction

Format SSF_DOLLARFR@(dDollar, fraction)

Arguments dollar_d A decimal number
fraction The integer to use as the fraction denominator.

Description SSF_DOLLARFR@ converts a dollar price expressed as a decimal number into a dollar price expressed as a fraction. Use SSF_DOLLARFR@ to convert decimal dollars into fractional dollars, such as prices for securities.

See also [SSF_DOLLARDE@](#).

SSF_DURATION@

Calculates the annual duration for a security with periodic interest payments

Format SSF_DURATION@(settlement, maturity, coupon, yld, frequency, basis)

Arguments All the arguments for this macro are described in the

Arguments	settlement	See Security Function Arguments
	maturity	See Security Function Arguments
	coupon	See Security Function Arguments
	yield	See Security Function Arguments
	frequency	See Security Function Arguments
	basis	See Security Function Arguments

Description SSF_DURATION@ calculates the annual duration for a security whose interest payments are made on a periodic basis. Duration is the weighted average of the present value of the bond's cash flows and is used as a measure of how a bond's price responds to changes in yield.

See also [SSF MDURATION@](#).

SSF_EFFECT@

Calculates the effective annual interest rate.

Format SSF_EFFECT@(nominal_rate,npery)

Arguments nominal rate Represents the nominal interest rate.
npery The number of compounding periods per year.

Description SSF_EFFECT@ calculates the effective annual interest rate. It is calculated using the following formula:

$$SSF_EFFECT@ = \left(1 + \frac{\text{nominal_rate}}{\text{npery}}\right)^{(\text{npery}-1)}$$

SSF_FVSCHEDULE@

Calculates the future value of an initial principal

Format SSF_FVSCHEDULE@(principal , schedule)

Description SSF_FVSCHEDULE@ calculates the future value of an initial principal after applying a series of compound interest rates. Use SSF_FVSCHEDULE@ to determine the future value of an investment with a variable or adjustable rate.

Arguments

principal	The present value.
schedule	An array of interest rates to apply. The values in schedule can be numbers or blank cells. Blank cells are treated as zeros (no interest).

SSF_INTRATE@

Calculates the interest rate for a fully invested security

Format SSF_INTRATE@(settlement, maturity, investment, redemption, basis)

Arguments

settlement	See Security Function Arguments
maturity	See Security Function Arguments
investment	See Security Function Arguments
redemptions	See Security Function Arguments
basis	See Security Function Arguments

Description SSF_INTRATE@ calculates the interest rate for a fully invested security. It is calculated using the following formula:

$$SSF_INTRATE = \frac{\text{redemption} - \text{investment}}{\text{investment}} \times \frac{B}{DIM}$$

where B is the number of days in a year, depending on the year basis, and DIM is the number of days from settlement to maturity.

See also [SSF_RECEIVED@](#).

SSF_MDURATION@

Calculates the annual modified duration for a security

Format SSF_MDURATION@(settlement, maturity, coupon, yld, frequency, basis)

Arguments

settlement	See Security Function Arguments .
maturity	See Security Function Arguments .
coupon	See Security Function Arguments .
yld	See Security Function Arguments .
frequency	See Security Function Arguments .
basis	See Security Function Arguments .

Description SSF_MDURATION@ calculates the annual modified duration for a security with interest payments made on a periodic basis, adjusted for market yield per number of coupon payments per year.

SSF_MDURATION@ uses the following formula to compute the modified duration:

$$SSF_MDURATION@ = \frac{SSF_DURATION@}{1 + \left(\frac{yld}{frequency} \right)}$$

See also [SSF_DURATION@](#)

SSF_NOMINAL@

Calculates the nominal annual interest rate

Format SSF_NOMINAL@(effect_rate,npery)

Arguments

effect_rate	The effective interest rate.
npery	The number of compounding periods per year, truncated to an integer.

Description SSF_NOMINAL@ calculates the nominal annual interest rate given the effective rate and the number of compounding periods per year. SSF_NOMINAL@ is calculated relative to SSF_EFFECT@ as shown in the following formula:

$$SSF_EFFECT@ = \left(1 + \frac{nominal_Rate}{nper}\right)^{(nper-1)}$$

See also [SSF_EFFECT@](#).

SSF_ODDFPRICE@

Calculates the price per \$100 face value of a security with an odd first period

Format SSF_ODDFPRICE@(settlement, maturity, issue, first_coupon, rate, yld, redemption, frequency, basis)

Arguments

settlement	See Security Function Arguments .
maturity	See Security Function Arguments .
issue	The security's issue date expressed as a serial date number.
first_coupon	The security's first coupon date.
rate	See Security Function Arguments .
yld	See Security Function Arguments .
redemption	See Security Function Arguments .
frequency	See Security Function Arguments .
basis	See Security Function Arguments .

Description SSF_ODDFPRICE@ calculates the price per \$100 face value of a security with an odd (short or long) first period. The dates must be related as follows:
 maturity > first_coupon > settlement > issue

SSF_ODDFPRICE@ is calculated using the following formulas:

For an odd short first coupon, SSF_ODDFPRICE@ is calculated using the following formula:

$$\begin{aligned}
 \text{SSF_ODDFPRICE@} = & \left[\frac{\text{redemption}}{\left(1 + \frac{\text{yld}}{\text{frequency}}\right)^{\left(N - 1 + \frac{\text{DSC}}{E}\right)}} \right] + \left[\frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{\text{DFC}}{E}}{\left(1 + \frac{\text{yld}}{\text{frequency}}\right)^{\frac{\text{DSC}}{E}}} \right] \\
 & + \left[\sum_{k=2}^N \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{\left(1 + \frac{\text{yld}}{\text{frequency}}\right)^{\left(k - 1 + \frac{\text{DSC}}{E}\right)}} \right] \\
 & - \left[100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{A}{E} \right]
 \end{aligned}$$

where *A* is the number of days from the beginning of the coupon period to the settlement date (accrued days);

DSC is the number of days from the settlement to the next coupon date;

DFC is the number of days from the beginning of the odd first coupon to the first coupon date;

E is the number of days in the coupon period;

N is the number of coupons payable between the settlement date and the redemption date. (If this number contains a fraction, it is raised to the next whole number.)

For an odd long first coupon, SSF_ODDFPRICE@ is calculated using the following formula:

$$\begin{aligned}
\text{SSF_ODDFPRICE@} = & \left[\frac{\text{redemption}}{\left(1 + \frac{\text{yld}}{\text{frequency}}\right)^{\left(N + N_q + \frac{\text{DSC}}{E}\right)}} \right] + \left[\frac{100 \times \frac{\text{rate}}{\text{frequency}} \times \sum_{i=1}^{NC} \frac{\text{DC}_i}{\text{NL}_i}}{\left(1 + \frac{\text{yld}}{\text{frequency}}\right)^{\left(N_q + \frac{\text{DSC}}{E}\right)}} \right] \\
& + \left[\sum_{k=2}^N \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{\left(1 + \frac{\text{yld}}{\text{frequency}}\right)^{\left(k - N_q + \frac{\text{DSC}}{E}\right)}} \right] \\
& - \left[100 \times \frac{\text{rate}}{\text{frequency}} \times \sum_{i=1}^{NC} \frac{A_i}{\text{NL}_i} \right]
\end{aligned}$$

where A_i is the number of days from the beginning of the i th quasi-coupon period within the odd period;

DC_i is the number of days from the dated date (or issue date) to the first quasi-coupon ($i = 1$) or the number of days in the quasi-coupon ($i = 2, \dots, i = \text{NC}$);

DSC is the number of days from the settlement to the next coupon date;

E is the number of days in the coupon period;

N is the number of coupons payable between the settlement date and the redemption date (if this number contains a fraction, it is raised to the next whole number);

NC is the number of quasi-coupon periods that fit in the odd period;

NL_i is the normal length in days of the full i th quasi-coupon period within the odd period;

N_q is the number of whole quasi-coupon periods between the settlement date and the first coupon.

SSF_ODDFYIELD@

Calculates the yield of a security with an odd first period

Format SSF_ODDFYIELD@(settlement, maturity, issue, first_coupon, rate, pr, redemption, frequency, basis)

Arguments

settlement	See Security Function Arguments .
maturity	See Security Function Arguments .
issue	The security's issue date, expressed as a serial date number.
first_coupon	The security's first coupon date. The dates must be related as follows: maturity > first_coupon > settlement > issue
rate	See Security Function Arguments .
pr	See Security Function Arguments .
redemption	See Security Function Arguments .
frequency	See Security Function Arguments .
basis	See Security Function Arguments .

Description SSF_ODDFYIELD@ calculates the yield of a security with an odd (short or long) first period.
SSF_ODDFYIELD@ is calculated using the Newton method, an iterative technique, based on the formula used for SSF_ODDFPRICE@. The yield is changed through 100 iterations until the estimated price having the given yield is near the price.

SSF_ODDLPRICE@

Calculates the price of a security with an odd last period

Format SSF_ODDLPRICE@(settlement, maturity, last_coupon, rate, yld, redemption, frequency, basis)

Arguments

settlement	See Security Function Arguments .
maturity	See Security Function Arguments .
last_coupon	The security's last coupon date. The dates must be related as follows: maturity > settlement > last_interest
rate	See Security Function Arguments .

yld	See Security Function Arguments .
redemption	See Security Function Arguments .
frequency	See Security Function Arguments .
basis	See Security Function Arguments .

Description SSF_ODDLPRICE@ calculates the price per \$100 face value of a security with an odd (short or long) last period.

SSF_ODDLYIELD@

Calculates the yield of a security with an odd last period.

Format SSF_ODDLYIELD@(settlement, maturity, last_coupon, rate, pr, redemption, frequency, basis)

Arguments	settlement	See Security Function Arguments .
	maturity	See Security Function Arguments .
	last_coupon	The security's last coupon date. The dates must be related as follows: maturity > settlement > last_interest See "Securities Function Arguments," earlier in this chapter, for definitions of the standard arguments for this function.
	rate	See Security Function Arguments .
	pr	See Security Function Arguments .
	redemption	See Security Function Arguments .
	frequency	See Security Function Arguments .
	basis	See Security Function Arguments .

Description SSF_ODDLYIELD@ calculates the yield of a security with an odd (short or long) last period. SSF_ODDLYIELD@ is calculated using the following formula:

SSF_ODDLYIELD@ =

$$\left[\frac{\left(\text{redemption} + \left(\sum_{i=1}^{NC} \frac{DC_i}{NL_i} \right) \times \frac{100 \times \text{rate}}{\text{frequency}} \right) - \left(\text{par} + \left(\sum_{i=1}^{NC} \frac{A_i}{NL_i} \right) \times \frac{100 \times \text{rate}}{\text{frequency}} \right)}{\text{par} + \left(\sum_{i=1}^{NC} \frac{A_i}{NL_i} \right) \times \frac{100 \times \text{rate}}{\text{frequency}}} \right] \times \left[\frac{\text{frequency}}{\left(\sum_{i=1}^{NC} \frac{DSC_i}{NL_i} \right)} \right]$$

where A_i is the number of accrued days for the i th quasi-coupon period within the odd period, counting forward from the last coupon date before redemption;

DC_i is the number of days counted in each i th quasi-coupon period as delimited by the length of the actual coupon period;

NC is the number of quasi-coupon periods that fit in the odd period (if this number contains a fraction it will be raised to the next whole number);

NL_i is the normal length in days of the full i th quasi-coupon period within the odd coupon period.

SSF_PRICE@

Calculates the price per \$100 of face value of a security that pays interest on a periodic basis.

Format SSF_PRICE@(settlement, maturity, rate, yld, redemption, frequency, basis)

Arguments settlement See [Security Function Arguments](#).

maturity See [Security Function Arguments](#).
 yld See [Security Function Arguments](#).
 redemption See [Security Function Arguments](#).
 frequency See [Security Function Arguments](#).
 basis See [Security Function Arguments](#).

settlement must not be greater than maturity.

Description SSF_PRICE@ calculates the price per \$100 of face value of a security that pays interest on a periodic basis.

SSF_PRICE@ is calculated using the following formula:

$$\text{SSF_PRICE@} = \left[\frac{\text{redemption}}{\left(1 + \frac{\text{yield}}{\text{frequency}}\right)^{\left(N - 1 + \frac{\text{DSC}}{E}\right)}} \right] + \left[\sum_{k=1}^N \frac{100 \times \frac{\text{rate}}{\text{frequency}}}{\left(1 + \frac{\text{yld}}{\text{frequency}}\right)^{\left(k - 1 + \frac{\text{DSC}}{E}\right)}} \right] - \left(100 \times \frac{\text{rate}}{\text{frequency}} \times \frac{A}{E} \right)$$

where *DSC* is the number of days from the settlement date to the next coupon date;

E is the number of days in the coupon period in which the settlement date falls;

N is the number of coupons payable between the settlement date and the redemption date;

A is the number of days from the beginning of the coupon period to the settlement date.

SSF_PRICEDISC@

Returns the price per \$100 of face value of a security that is discounted instead of paying periodic interest

Format SSF_PRICEDISC@(settlement, maturity, discount, redemption, basis)

Arguments

settlement	See Security Function Arguments .
maturity	See Security Function Arguments .
discount	See Security Function Arguments .
redemption	See Security Function Arguments .
frequency	See Security Function Arguments .
basis	See Security Function Arguments .

The argument settlement must not be greater than maturity.

Description SSF_PRICEDISC@ returns the price per \$100 of face value of a security that is discounted instead of paying periodic interest.
SSF_PRICEDISC@ is calculated using the following formula:

$$\text{SSF_PRICEDISC@} = \text{redemption} - \text{discount} \times \frac{DSM}{B}$$

where B is the number of days in the year, depending on the year basis used; DSM is the number of days from settlement to maturity.

SSF_PRICEMAT@

Calculates the price per \$100 of face value of a security that pays its interest at the maturity date.

Format SSF_PRICEMAT@(settlement, maturity, issue, rate, yld, basis)

Arguments

settlement	See Security Function Arguments
maturity	See Security Function Arguments
issue	See Security Function Arguments
rate	See Security Function Arguments
yld	See Security Function Arguments

basis See [Security Function Arguments](#)

The argument settlement must not be greater than maturity.

Description SSF_PRICEMAT@ calculates the price per \$100 of face value of a security that pays its interest at the maturity date.

SSF_PRICEMAT@ is calculated using the following formula:

$$\text{SSF_PRICEMAT@} = \frac{100 + \left(\frac{DIM}{B} \times rate \times 100 \right)}{1 + \left(\frac{DSM}{B} \times yld \right)} - \left(\frac{A}{B} \times rate \times 100 \right)$$

where B is the number of days in the year, depending on the year basis used; DSM is the number of days from settlement to maturity; DIM is the number of days from issue to maturity; A is the number of days from issue to settlement.

SSF_RECEIVED@

Calculates the amount received at maturity from a fully invested security

Format SSF_RECEIVED@(settlement, maturity, investment, discount, basis)

Arguments settlement See [Security Function Arguments](#).

maturity See [Security Function Arguments](#).

investment See [Security Function Arguments](#).

discount See [Security Function Arguments](#).

basis See [Security Function Arguments](#).

The argument settlement must not be greater than maturity.

Description SSF_RECEIVED@ calculates the amount received at maturity for a fully invested security. SSF_RECEIVED@ is calculated using the following formula:

$$\text{SSF_RECEIVED@} = \frac{investment}{1 - \left(discount \times \frac{DIM}{B} \right)}$$

where B is the number of days in the year, depending on the year basis used; DIM is the number of days from issue to maturity.

SSF_TBILLEQ@

Calculates the bond-equivalent yield for a Treasury bill

Format SSF_TBILLEQ@(settlement, maturity, discount)

Arguments settlement See [Security Function Arguments](#).
maturity See [Security Function Arguments](#).
discount See [Security Function Arguments](#).

Description SSF_TBILLEQ@ calculates the bond-equivalent yield for a Treasury bill. The argument settlement must not be greater than maturity, and maturity may not be more than one year after settlement.
SSF_TBILLEQ@ is calculated using the following formula:

$$\text{SSF_TBILLEQ@} = \frac{365 \times \text{discount}}{360 - (\text{discount} \times \text{DSM})}$$

where DSM is the number of days from settlement to maturity computed according to the 360 days per year basis.

SSF_TBILLPRICE@

Calculates the price per \$100 of face value of a Treasury bill

Format SSF_TBILLPRICE@(settlement, maturity, discount)

Arguments settlement See [Security Function Arguments](#).
maturity See [Security Function Arguments](#).
discount See [Security Function Arguments](#).

The argument settlement must not be greater than maturity, and maturity may not be more than one year after settlement.

Description SSF_TBILLPRICE@ calculates the price per \$100 of face value for a Treasury bill.
SSF_TBILLPRICE@ is calculated using the following formula:

$$\text{SSF_TBILLPRICE@} = 100 \times \left(1 - \frac{\text{discount} \times \text{DSM}}{360} \right)$$

where DSM is the number of days from settlement to maturity, excluding any maturity date that is more than one calendar year after the settlement date.

SSF_TBILLYIELD@

Calculates a Treasury bill's yield

Format SSF_TBILLYIELD@(settlement, maturity, par)

Arguments settlement See [Security Function Arguments](#).
 maturity See [Security Function Arguments](#).
 par See [Security Function Arguments](#).

The argument settlement must not be greater than maturity, and maturity may not be more than one year after settlement.

Description SSF_TBILLYIELD@ calculates a Treasury bill's yield. SSF_TBILLYIELD@ is calculated using the following formula:

$$\text{SSF_TBILLYIELD@} = \frac{100 - \text{par}}{\text{par}} \times \frac{360}{\text{DSM}}$$

where DSM is the number of days from settlement to maturity, excluding any maturity date that is more than one calendar year after the settlement date.

SSF_XIRR@

Calculates the internal rate of return for a schedule of cash flows that is not necessarily periodic

Format SSF_XIRR@(values, dates, guess)

Arguments values A series of cash flows that correspond to the schedule of payments listed in the dates argument. The first payment is optional, and corresponds to a

cost or payment at the beginning of the investment. All subsequent payments are discounted based on a 365-day year.

- dates A schedule of payment dates that corresponds to the cash flow payments in values. The first payment date is the beginning of the schedule of payments. All the other dates must be later than the beginning date, but they may occur in any order.
- guess A number you estimate as the result of XIRR. In most cases you can omit guess. If omitted, guess is assumed to be 0.1 or 10%.
 SSF_XIRR@ expects at least one positive cash flow and one negative cash flow. The values and dates arguments must contain the same number of items in the lists.

Description SSF_XIRR@ calculates the internal rate of return for a schedule of cash flows that is not necessarily periodic.

SSF_XIRR@ is closely related to the function SSF_XNPV@, or net present value. The rate of return calculated by SSF_XIRR@ is the interest rate that corresponds to SSF_XNPV@ = 0.

SSF_XIRR@ uses an iterative calculation technique. Using a changing rate (starting with guess), SSF_XIRR@ calculates repeatedly until the result is accurate within 0.000001%. If the calculation cannot find a sufficiently accurate result after 100 iterations, SSF_XIRR@ returns an error message.

The iterative calculation used for SSF_XIRR@ changes the rate until:

$$0 = \sum_{i=1}^N \left(\frac{P_i}{\left(1 + \text{SSF_XIRR@}\right)^{\frac{(d_i - d_0)}{365}}} \right)$$

where d_j is the i th payment date; d_0 is the 0th payment date, at the beginning of the investment; P_i is the i th payment.

SSF_XNPV@

Calculates the net present value for a schedule of cash flows that is not necessarily periodic

Format SSF_XNPV@(rate, values, dates)

Arguments

rate	The discount rate to apply to the cash flows.
values	A series of cash flows that correspond to the schedule of payments listed in the dates argument. The first payment is optional, and corresponds to a cost or payment at the beginning of the investment. All subsequent payments are discounted based on a 365-day year.
dates	A schedule of payment dates that corresponds to the cash flow payments in values. The first payment date is the beginning of the schedule of payments. All the other dates must be later than the beginning date, but they may occur in any order.

The values and dates arguments must contain the same number of items in the lists.

Description SSF_XNPV@ calculates the net present value for a schedule of cash flows that is not necessarily periodic.

SSF_XNPV@ is calculated using the following formula:

$$\text{XNPV} = \sum_{i=1}^N \left(\frac{P_i}{(1 + \text{rate})^{\frac{(d_i - d_0)}{365}}} \right)$$

where d_j is the j th payment date; d_0 is the 0th payment date, at the beginning of the investment; P_j is the j th payment.

SSF_YIELD@

Calculates the annual yield for a security that pays interest on a periodic basis

Format SSF_YIELD@(settlement, maturity, rate, pr, redemption, frequency, basis)

Arguments	settlement	See Security Function Arguments .
	maturity	See Security Function Arguments
	rate	See Security Function Arguments
	pr	See Security Function Arguments
	redemption	See Security Function Arguments
	frequency	See Security Function Arguments
	basis	See Security Function Arguments

The argument settlement must not be greater than maturity.

Description SSF_YIELD@ calculates the annual yield for a security that pays interest on a periodic basis.

If there is one coupon period or less until redemption, SSF_YIELD@ is calculated using the following formula:

$$\text{SSF_YIELD@} = \frac{\left(\frac{\text{redemption}}{100} + \frac{\text{rate}}{\text{frequency}} \right) - \left(\frac{\text{pr}}{100} + \left(\frac{A}{E} \right) \times \frac{\text{rate}}{\text{frequency}} \right)}{\frac{\text{pr}}{100} + \left(\frac{A}{E} \times \frac{\text{rate}}{\text{frequency}} \right)} \times \frac{\text{frequency} \times E}{\text{DSR}}$$

where A is the number of days from the beginning of the coupon period to the settlement date (accrued days); DSR is the number of days from settlement to redemption; and E is the number of days in the coupon period.

If there is more than one coupon period before redemption, SSF_YIELD@ is calculated through 100 iterations, using the Newton method based on the formula for SSF_PRICE@, changing the yield until the estimated price given the yield is close to the actual price.

SSF_YIELDDISC@

Calculates the annual yield for a discounted security

Format SSF_YIELDDISC@ (settlement, maturity, pr, redemption, basis)

Arguments

settlement	See Security Function Arguments .
maturity	See Security Function Arguments .
pr	See Security Function Arguments .
redemption	See Security Function Arguments .
basis	See Security Function Arguments .

Description SSF_YIELDDISC@ calculates the annual yield for a discounted security. SSF_YIELDDISC@ is calculated iteratively using SSF_PRICEDISC@. The argument settlement must not be greater than maturity.

SSF_YIELDMAT@

Calculates the annual yield for a security that pays interest at maturity

Format SSF_YIELDMAT@(settlement, maturity, issue, rate, pr, basis)

Arguments

settlement	See Security Function Arguments
maturity	See Security Function Arguments
issue	See Security Function Arguments
rate	See Security Function Arguments
pr	See Security Function Arguments
basis	See Security Function Arguments

The argument settlement must not be greater than maturity.

Description SSF_YIELDMAT@ calculates the annual yield for a security that pays interest at maturity. SSF_YIELDMAT@ is calculated iteratively using SSF_PRICE_MAT@.

Security Function Arguments

The financial functions used to calculate and analyze securities share many similar arguments, including the following:

basis Day count basis of the security, as shown in the following table:

Argument	Type	Description
0 or omitted	US (NASD) 30/360 basis	Considers each month to have 30 days, and each year to have 360 days.
1	actual/actual	Considers each month to have the actual number of calendar days in that month, and each year to have the actual number of calendar days in that year.
2	actual/360	Considers each month to have the actual number of calendar days in that month, and each year to have 360 days.
3	actual/365	Considers each month to have the actual number of calendar days in that month, and each year to have 365 days.
4	European 30/360	Considers each month to have 30 days, and each year to have 360 days, according to the European system.

coupon Annual coupon interest rate of the security.

discount The security's discount rate.

frequency Number of coupon payments made per year:

1 annual

2 semiannual

4 quarterly

investment The initial purchase price of a security.

maturity Maturity date of the security, expressed as a serial date number.

pr	The security's price per \$100 face value.
rate	Interest rate of the security at issue date.
redemption	Value of the security at redemption.
settlement	Settlement date of the security (the date it must be paid for), expressed as a serial date number.
yld	Annual yield of the security.

To enter date arguments, use the **DATE@** function to convert a calendar date to a serial date number.

For more information on securities and their calculations, refer to industry standard publications such as *Standard Securities Calculation Methods*, published by the Securities Industry Association (Third Edition, © 1993).

SS_GET_BORDERS@

Returns the cell border attributes

Format format borders_info_info = SS_GET_BORDERS@(cellOrRange)

Method format borders_info_info = this.get_borders@(cellOrRange)

Arguments cellOrRange A string indicating a cell address or a spreadsheet range.

Description Returns the cell border attributes for the current selection in a Spreadsheet. This format is defined as follows:

```
format borders_info_
  format ss_line_attrs outline,
  format ss_line_attrs top,
    Same values as outline
  format ss_line_attrs bot,
    Same values as outline
  format ss_line_attrs left,
    Same values as outline
  format ss_line_attrs right,
  format ss_shade attrs shading
```

The definition of ss_line_attrs is as follows:

```
format ss_line_attrs
  style,  -1  mixed
          0  off
```

1	thin
2	medium
3	thick
4	dashed
5	double
color	The color's string name

The definition of `ss_shade_attrs` is as follows:

```
format ss_shade_attrs
  style, 0 to 19: These numbers (which represent the shading patterns) can be
             derived from the shade palette, reading left to right, top to bottom.
  fgcolor, The color's string name
  bgcolor The color's string name
```

`SS_GET_BORDERS@` is called by the Style ® Borders menu option.

See also [SS SET BORDERS@](#)

SS_GET_CALC_OPTIONS@

Returns information on how the spreadsheet 'calc' is performed

Format `format ss_calc_options@ info = SS_GET_CALC_OPTIONS@()`

Method `format ss_calc_options@ info = this.get_calc_options@`

Description Returns a variable containing information on how the spreadsheet recalculates formulas. This information is returned in a `ss_calc_options@` format whose definition is as follows:

```
format ss_calc_options@
  mode,
  style,
  iteration_count,
  calc_interval,
  auto_chart,
  calc_background,
  calc_on_display,
  calc_rtinsert_on_display,
  calc_only_obsolete_cells,
  type_conversion
```

SS_GET_CELL@

Returns information about a cell

Format format ss_cell_ cellinfo = SS_GET_CELL@(col, row[, sheet])

Method format ss_cell_ cellinfo = this.get_cell@(col, row[, sheet])

Arguments	col	The column containing the cell from which you want information. col must be a numeric value. Each spreadsheet column is assigned a number, beginning with 0. Thus, column A is 0, column B is 1, and so on. Column ZZ is 701.
	row	The row containing the cell from which you want information. row is a numeric value, with the first row being row 0. The last row is 32,766.
	sheet	The sheet containing the cell from which you want information. sheet must be a numeric value. Each spreadsheet sheet is assigned a number, beginning with 0. Thus, sheet A is 0, sheet B is 1, and so on. If you omit this value, the value of the current sheet is used.

Description Returns a ss_cell_ format containing information about a cell's attributes.

The definition of ss_cell_ and its contents is as follows:

format ss_cell_	
row,	The row number (0 - 32,766) of the specified cell.
col,	The column number (0 - 701) of the specified cell.
type,	The cell type. The value can be one of the following:
	100 SSC#CELL_NUM_FORMULA Cell contains a valid formula
	101 SSC#CELL_TEXT_FORMULA Cell contains a string formula
	102 SSC#CELL_BOOL_FORMULA
	103 SSC#CELL_OBSOLETE Cell is obsolete; the cell value needs to be recalculated
	104 SSC#CELL_ERROR Cell has the value ERROR
	105 SSC#CELL_NA Cell has the value NA
	106 SSC#CELL_PENDING Cell is empty
	107 SSC#CELL_CIRCULAR Cell is part of a circular reference
	108 SSC#CELL_DIV_ZERO

109 SSC#CELL_UDEFNAME
 110 SSC#CELL_NUMERR
 111 SSC#CELL_TYPERR
 112 SSC#CELL_REFERR
 113 SSC#CELL_ARGERR
 114 SSC#CELL_FAILERR
 115 SSC#CELL_DBERR1
 116 SSC#CELL_DBERR2
 117 SSC#CELL_DBERR3
 200 SSC#CELL_NUMERIC
 Cell contains a numeric constant
 201 SSC#CELL_TEXT
 Cell contains a label
 204 SSC#CELL_EMPTY
 Cell is empty
 205 SSC#CELL_NA_VAL
 206 SSC#CELL_ERROR_VAL

display_str, The cell's display string, as it appears in the cell. This string reflects the results of any computation or number style in effect. For example, if this cell contains a formula, `display_str` displays the result of that computation, with the precision defined for in that cell.

entry_str, The actual entry that was in this cell, as it appeared in the edit line before the latest entry. If the cell had a formula, this element tells you what the formula was rather than the computation of that formula. Similarly, a number appears here as it existed before any results of the `style_type` modifications.

style_type, The style attribute assigned to the cell. `style_type` is a string made up of one or more of the following codes. The string ends with a `|' (pipe) delimiter followed by any border attributes.

Protection, visibility:

P protected
 I invisible

Number formats:

Gf unstyled
 C# currency (# = precision)
 CO# comma (# = precision)
 F# fixed (# = precision)
 G# general (# = precision)
 S# scientific notation (# = precision)
 P# percentage (# = precision)

Date formats:

B0 Boolean
GR graph
D1 date format mm/dd/yy
D2 date format mmm dd, yyyy
D3 date format dd.mm.yy
D4 date format dd mmm yy
D5 date format month dd, yyyy
D6 date format dd.mm.yyyy

Time formats:

T0 time format hh:mm:ss AM/PM
T1 time format hh:mm AM/PM
T2 time format hh:mm:ss
T3 time format hh:mm

Date and time formats are defined in the file

`/<install_dir>/<lang>/datetime_.sp` where *install_dir* is the Ap-
plixware installation directory, and *lang* is eng (english), frn (french),
or grm (german). If `datetime_.sp` has been modified at your site,
your date and time formats may differ from this list.

Horizontal Justification:

1 left justified
2 right justified
3 centered
4 repeat

Vertical Justification:

VB Bottom justified
VT Top justified
VC centered

Underlining:

B Bold
I Italic
U Single underline
D Double underline

Typefaces:

TF n Type face where n is a number starting at 1 that is incre-
mented each time the type face of the cell is changed, un-
less the cell is changed to a typeface that has previously
been selected.

For example, suppose when you first start the spreadsheet,
Cell A:A1 is Helvetica. The value in this field is TF1.

You change the cell to Palatino. The value of this field becomes TF2.

You then change the cell to Zapf Dingbats. The value of this cell becomes TF3.

You change the cell back to Helvetica. The value of the cell becomes TF1.

Contents Color:

FGn *n* indicates the color in the palette that is selected

Point Size:

Pn Point size of the cell text

Wrap text:

WT Indicates that text wrap is enabled

Border attribute(s):

| (pipe)

A delimiter that separates any of the above attributes from the border attributes of the cell. If the cell has no borders drawn around it, then the *style_type* string ends with the | delimiter. Border attributes are separated by a comma (as in the outline-border example).

SHn Shade Type

FGn Shade Foreground Color

BGn Shade Background Color

TnFGn

Top border line type and color

BnFGn

Bottom border line type and color

LnFGn

Left border line type and color

RnFGn

Right border line type and color

HJust

Number indicating the horizontal justification, as follows:

1 left justified

2 right justified

3 centered

4 repeat

wrap_text

TRUE if *wrap_text* justification is on, valid only for textcells

protected

true if protected, false if not protected

invisible	true if invisible, false if visible
precision	extent of display style
value	current value if cell is numeric
sheet	zero based sheet number
valign	Type of vertical alignment

SS_GET_CELL_ATTR_INFO@

Returns information on the formatting of a range of cells

Format format ss_cell_attr_info@ info = ss_get_cell_attr_info@(range, borderFlag)

Arguments range A string containing a range of cells, such as A:B4..A:C7.
borderFlag If TRUE, the macro returns border information. If FALSE, no border information is returned.

Description Returns an array of information about the format attributes of a range of cells. The array is an ss_cell_attr_info@ format. The ss_cell_attr_info@ format, and the formats contained in it, are described in the file [spsheet.am](#).

SS_GET_CALC_CELL@

Return the cell whose formula is currently being calculated

Format cell = SS_GET_CALC_CELL@()

Description Returns the cell address that is currently being calculated.

SS_GET_CELL_DATA@

Returns the value of the specified cell

Format value = SS_GET_CELL_DATA@(cellAddr)

Method value = this.get_cell_data@(cellAddr)

Arguments cellAddr A cell address. This is either an array of 3 elements containing a column, row and sheet, or a string such as "A:A4".

Description Returns a cell's contents as follows:

- If the cell contains a label, the label string is returned.
- If the cell contains a string or a formula that returns a string, the displayed string is returned.
- If the cell contains an ERROR or NA datum, a binary object is returned.
- If the cell contains a number or a formula that returns a number, the displayed number is returned.

SS_GET_CELL_VALUE@

Returns the value of the specified cell

Format value = SS_GET_CELL_VALUE@(col, row[, sheet])

Method value = this.get_cell_value@(col, row[, sheet])

Arguments

col	The column number of the cell. Column numbers are zero-based, with column A being 0, column B being 1, and so on.
row	The row number of the cell. Row numbers are 0-based, with row 1 being 0, row 2 being 1, and so on.
sheet	The sheet number of the cell. Sheet numbers are 0-based, with sheet A being 0, sheet B being 1, and so on. If this argument is omitted, the default sheet is the current sheet.

Description Returns a cell's contents as follows:

- If the cell contains a label, the label string is returned.
- If the cell contains a string formula, the display string is returned.
- If the cell contains a number or a formula, the number is returned.
- In all other cases, the display string is returned.

SS_GET_CELL_STATS@

Returns Cell Statistics

Format SS_GET_CELL_STATS@()

Description Returns an SS_CELL_STATISTICS@ format that describes the contents of the current spreadsheet. The SS_CELL_STATISTICS@ format contains the following values:

format ss_cell_statistics@

num_cells,	'total number of cells containing a value
num_numeric_cells,	'total number of numeric cells
num_text_cells,	'total number of text cells
num_empty_cells,	'total number of empty cells
num_formula_cells,	'total number of formula cells
num_rti_cells,	'total number of RTINSERT cells
num_rts_cells,	'total number of RTS cells
num_non_rt_cells,	'total number of non-Real Time formula cells
format arrayof ss_cell_loc@ cell_bound	
	' array containing the last row, last column
	' and sheet number for each sheet in the
	' spreadsheet.

All numbers returned by SS_GET_CELL_STATS@, such as the number of cells or the sheet number, are zero-based numbers.

SS_GET_CHARTS_IN_DOC@

Returns a list of charts

Format nameArray = SS_GET_CHARTS_IN_DOC@(docName)

Method nameArray = this.get_charts_in_doc@(docName)

Arguments docName The name of a Spreadsheets document.

Description Returns an array containing the names of all charts within a Spreadsheets document.

SS_GET_COLOR_TABLE@

Returns an array of color information

Format format arrayof ss_color@ color = SS_GET_COLOR_TABLE@()

Method format arrayof ss_color@ color = this.get_color_table@

Description Returns an array of color information. Each element of the returned array is one color table entry. These entries are defined using the ss_color@ format, whose definition is as follows:

format ss_color@
name,
cyan,
magenta,
yellow,
black

SS_GET_COL_WIDTH@

Returns a column's width

Format width = SS_GET_COL_WIDTH@(colStr)

Method width = this.get_col_width@(colStr)

Arguments colStr The string name of a column.

Description Returns a column's width. The value of the returned width indicates the number of characters that can be printed within the column.

This macro only works in the current sheet.

See also [SS_SET_COL_WIDTH@](#)
[SS_VIEW_GET_COL_WIDTH@](#)

SS_GET_CURRENCY@

Returns currency information

Format format ss_currency@ cur = SS_GET_CURRENCY@()

Method format ss_currency@ cur = this.get_currency@

Description Returns information on the way in which currency is displayed.

This information is returned as a ss_currency@ format, whose definition is as follows:

format ss_currency@	
currency_str,	` The currency string itself; for example, "\$"
european,	` TRUE if European, FALSE if English
trailing	` TRUE if trailing, FALSE if leading

See also [SS_SET_CURRENCY@](#)

SS_GET_DB_INFO@

Returns information describing an on-sheet database

Format infoArray = SS_GET_DB_INFO@(dbName)

Method infoArray = this.get_db_info@(dbName)

Arguments dbName The name of an onsheet database.

Description Returns a five element array containing information about an onsheet database, as follows:

4. The database name
5. The range string
6. The criterion range
7. The extraction range
8. The clear extract range string

See also [SS_SET_DATABASE@](#)

SS_GET_DB_NAMES@

Returns an array of onsheet database names

Format nameArray = SS_GET_DB_NAMES@()

Method nameArray = this.get_db_names@

Description Returns an array whose elements contain the names of onsheet databases.

SS_GET_DOC_ATTR@

Returns general status information

Format format ss_status_info = SS_GET_DOC_ATTR@()

Method format ss_status_info = this.get_doc_attr@

Description Writes many of general document attributes into a ss_status_ format. The definition of this format is as follows:

format ss_status_	
open_col,	'The column of the cell cursor
open_row,	'The row of the edit cursor
auto_calc,	'0 manual
	'1 auto
	'2 interval
calc_mode,	'0 normal
	'1 row
	'2 column
edit_mode,	'Boolean
point_mode,	'Not used
bottom_col,	'Last cell's column
bottom_row,	'Last cell's row
auto_graphing,	'Boolean: if TRUE, changes in data cause change
	'in graph
calc_count,	'1-10 (not used for
	'normal calc mode
font_size,	'the point size
print_size,	'Not used
text_align,	'0 left
	'1 right
	'2 center
num_align,	'0 left
	'1 right
	'2 center
	'3 repeat
num_style,	'0 unstyled
	'1 Boolean
	'2 general
	'3 fixed
	'4 scientific
	'5 money
	'6 comma
	'7 percent
	'8 date
	'9 graph
	'10 time
	'11 default
num_prec,	'0 - 9
bold,	'Boolean: TRUE if in bold state
face,	'String
underline,	'0 none
	'1 single

	'2	double
	'null	As is
italic,		'Boolean: TRUE if in italic state
color,		'String name of the color
gridstyle,		'One of the following
	'0	None
	'1	Dotted line
	'2	Solid line
wrap_text		'Boolean: TRUE if wrapping within cells
open_sheet,		'Current-displayed sheet (0 = sheet A)
bottom_sheet,		'Last sheet containing a value (0 = sheet A)
defcolwidth,		'Default Column width, as set through View ® Column Width
minimum_recalc,		' a boolean. 0 = Calculate obsolete cells
		1 = Calculate all cells
calc_interval,		Calculation Interval as set through Style ® Calculation and Chart Update

See also [SS SET DOC ATTR@](#)

SS_GET_DOC_INFO@

Returns an array of information on the current spreadsheet

Format format doc_format_info = SS_GET_DOC_INFO@()

Method format doc_format_info = this.get_doc_info@

Description Returns an array giving the following information on the current spreadsheet. The FORMAT template for the array information is doc_format_. The header file containing the FORMAT template is fileinf_.am.

format doc_format_	
name,	The full path name of the current spreadsheet.
docid,	The spreadsheet's unique ID number.
on_disk,	Indicates if the spreadsheet has been saved or read using the file specified by name. Returns TRUE if the spreadsheet has been saved or read, FALSE if it hasn't.
save_mode,	The mode in which the spreadsheet has been saved. save_mode is one of the following:
	0 binary
	1 normal
	2 DIF format
	3 SYLK format

grp_access, The permission setting for the file for members of the same group. grp_access is one of the following:

- 0 none
- 1 read
- 2 read/write

all_access, The permission setting for the file for all users. all_access is one of the following:

- 0 none
- 1 read
- 2 read/write

writable Indicates whether the spreadsheet file is writable or read-only. Returns TRUE if the user has write access to the file. Returns FALSE if the file is read-only.

SS_GET_FONT_FAMILIES@

Returns an array of font families available

Format fontArrayInfo = SS_GET_FONT_FAMILIES@()

Method fontArrayInfo = this.get_font_families@

Description Returns an array containing the fonts that can be used within a Spreadsheets.

SS_GET_FONT_SIZES@

Returns an array containing the text point sizes available

Format sizeArray = SS_GET_FONT_SIZES@()

Method sizeArray = this.get_font_sizes@

Description Returns an array containing the built-in (default) point sizes that can be used to display fonts.

SS_GET_GLOBALS@

Returns information about the current state of the Spreadsheet

Format SS_GET_GLOBALS@()

Description Returns an SS_GLOBALS_ array format. This format is defined in the file spsheet_.am.

See also [SS_GLOBALS format](#)

SS_GET_HDRFTR@

Returns header and footer information

Format format hdrftr_info info = SS_GET_HDRFTR@()

Method format hdrftr_info info = this.get_hdrftr@

Description Returns header and footer information.

The structure of the returned hdrftr_info format is as follows:

format hdrftr_info

special_even_off, A value indicating if odd and even pages have the same or different headers and/or footers.

0 headers are the same

1 headers differ

special_first, A value indicating if the first page's header and/or footer differ from the headers and footers on the remaining pages.

0 first page is not special

1 first page is special

special_final, A value indicating if the last page's header and/or footer differ from the headers and footers on the preceding pages.

0 last page is not special

1 last page is special

format hdrftr_pg_info base,

format hdrftr_pg_info even,

format hdrftr_pg_info odd,

format hdrftr_pg_info first,

format hdrftr_pg_info final

The structure of hdrftr_pg_info is as follows:

```

format hdrftr_pg_info
    hdr_offset,          The header offset. This is the amount of space reserved for
                        the header.
    ftr_offset,         The footer offset. This is the amount of space reserved for
                        the footer.
    format hdrftr_line_info hdr,
    format hdrftr_line_info ftr

```

If elements 3 or 4 (hdr and ftr) are arrays, each array element indicates one line in the header or footer. For example, if the dimension of hdr is 2 and the dimension of ftr is 3, then the spreadsheet has a two-line header and a three-line footer.

The structure of `hdrftr_line_info` is as follows:

```

format hdrftr_line_info
    format hdrftr_group_inner,
    format hdrftr_group_centr,
    format hdrftr_group_outer

```

The structure of `hdrftr_group_` is as follows:

```

format hdrftr_group_
    text,                'text of the header/footer segment
    face,                'the font
    size,                'point size
    color,               'color
    bold,                'Boolean
    italic,              'Boolean
    strike,              'Boolean
    underline,           'One of the following:
                        '0    none
                        '1    single
                        '2    double
    word_underline       'Boolean: TRUE if underlining is by word

```

The easiest way to create a header and footer for a document is to build one interactively. You could then use this macro to obtain the information and write it to disk. You could then apply this data to a new Applixware Spreadsheets document using [**SS SET HDRFTR@**](#).

SS_GET_HOOK@

Returns a hook macro's name

Format macroName = SS_GET_HOOK@()

Method macroName = this.get_hook@

Description Returns the name of the hook macro defined for a Spreadsheets document. If no macro is associated with the document, NULL is returned.

See also [SS_SET_HOOK@](#).

SS_GET_LINKS@

Returns an array indicating all the external links

Format linkArray = SS_GET_LINKS@()

Method linkArray = this.get_links@

Description Returns a string array in which each array element is the full path name of an external file that is linked to the current spreadsheet.

SS_GET_NAMED_RANGE_INFO@

Returns information about a named range

Format format named_range_format_info = SS_GET_NAMED_RANGE_INFO@ (rangeName)

Method format named_range_format_info = this.get_named_range_info@ (rangeName)

Arguments rangeName An existing range name. If name does not exist, an error is thrown.

Description Returns an array containing the following information about the named range. The FORMAT template name is named_range_format_ and is contained in spsheet_.am.

format named_range_format_

name, The name of the range. If the range is a link to an external range, the alias used for the range in the current spreadsheet is returned.

range, The cell address of the named range.

docname, This information is only returned for ranges that are links to external ranges. docname is the name of the spreadsheet file in which the external range is located.

docrange, This information is only returned for ranges that are links to external ranges. docrange is the name of the external range.

use_linked_attrs

This information is only returned for ranges that are links to external ranges. This is a Boolean value which if set to TRUE indicates that the attributes of the linked range should be used.

Description Returns information describing a range. The structure of this returned information is described above.

NOTE: If the cursor is below the cells that are ranges (for example, the range is A1 and the cursor location is in A3), the cursor is then moved to A1, the "relative range" will wrap to the bottom of the column.

See also [SS_NAME_EXTERNAL@](#)

SS_GET_NAMES@

Returns the names of all named ranges

Format nameArray = SS_GET_NAMES@(mode)

Method nameArray = this.get_names@(mode)

Arguments

mode	The type of names to be listed, as follows:
0	All
1	Local names
2	Link names

Description Returns an array in which each element of the array is the name of a named range in the current spreadsheet. If no named ranges are defined for the spreadsheet, NULL is returned.

See also [SS_NAME_CHANGE@](#)
[SS_NAME_CREATE@](#)
[SS_NAME_DELETE@](#)
[SS_NAME_EXTERNAL@](#)

SS_GET_NAMES_IN_DOC@

Returns a list of named objects

Format nameArray = SS_GET_NAMES_IN_DOC@(docname)

Method nameArray = this.get_names_in_doc@(docname)

Arguments docname The document's path name.

Description Returns a list of named objects contained within a document. This list does not include the names of any charts.

SS_GET_OBJ_INFO@

Returns object property information

Format format ss_obj_info@ = SS_GET_OBJ_INFO@(objectname)

Method format ss_obj_info@ = this.get_obj_info@(objectname)

Arguments objectname The name of an object embedded within or linked to the current Spreadsheets document.

Description Returns an ss_obj_info@ format whose contents describe the properties of objectname. The definition of this format is as follows:

```
format ss_obj_info@
    name,          'string
    type,          'integer
    property,      'Boolean
    hidden,        'Boolean
    locked,        'Boolean
    extlink,       'Boolean
    print,         'integer
    path,          'string; only used if extlink is TRUE
    macro_to_run, 'string: name of elf macro
    no_border     'Boolean
```

SS_GET_OBJECTS@

Gets the current list of a certain type of object

Format arrayOfObjects = SS_GET_OBJECTS@(type, needUpdateFlag)

Arguments type The type of object to list
needUpdateFlag Forces an update to the objects in the chart before returning information.

Description Lists all of the objects in the current spreadsheet that match the requested type. The spreadsheet object types are defined the file SPSHEET_.AM as follows:

Define Name	Hexadecimal value
SSOBJ#GRAPHIC- S_	0x01
SSOBJ#CHART_	0x02
SSOBJ#EQUATIO- N_	0x04
SSOBJ#AUDIO_	0x08
SSOBJ#BUTTON_	0x10
SSOBJ#ALL	0x1F

This macro returns an array. Each element in the array is an SS_OBJ_INFO@ format. SS_OBJ_INFO@ is defined in the ELF include file [spsheet .am](#).

SS_GET_PAGEBREAKS@

Gets a list of page breaks

Format breakstring = SS_GET_PAGEBREAKS@([viewname])

Arguments viewname The name of a named view in the current spreadsheet.

Description Returns a comma-delimited string containing the rows and columns in the current spreadsheet that have page breaks installed. The rows are listed first in the string, then the columns. For example, if you have page breaks in your spreadsheet at rows 4 and 21, and columns C and G, the returned string is:

4, 21, C, G

Viewname is an optional parameter. If it is omitted, the page breaks in the current view are returned.

See also [SS_SET_PAGEBREAKS@](#)
[SS_CLEAR_PAGEBREAKS@](#)

SS_GET_PAGE_SETUP@

Returns print setup attributes

Format format ss_page_setup_info = SS_GET_PAGE_SETUP@()

Description Returns the following print setup information for the current spreadsheet. The FORMAT template for SS_GET_PAGE_SETUP@ is named ss_page_setup_ and is contained in spsheet_.am. The definition of this format is as follows:

```
format ss_page_setup_
width,          The width of the document.
height,        The height of the document.
lmargin,       The document's left printing margin.
rmargin,       The document's right printing margin.
tmargin,       The document's top printing margin.
bmargin,       The document's bottom printing margin.
landscape,     Indicates the orientation of the document. Returns 0 for portrait ori-
               entation, 1 for landscape orientation.
center_halign, Indicates if the spreadsheet is centered from left to right on the
               printed page.
center_valign, Indicates if the spreadsheet is centered from top to bottom on the
               printed page.
prt_headers,   A Boolean value which if set to TRUE indicates that row and col-
               umn headers should be printed.
facing_pages, A Boolean value which if set to TRUE indicates that left and right
               pages will be used.
print_to_fit,  A Boolean value which if set to TRUE indicates that Spreadsheets
               should override the current page break settings and print the docu-
               ment in the number of pages that are indicated in num_pages_wide
               and num_pages_tall.
num_pages_wide, The number of horizontal pages to be used when printing a docu-
               ment.
num_pages_tall, The number of vertical pages to be used when printing a document.
no_print_beyond_last_cell
               A Boolean value which if set to TRUE indicates that Spreadsheets
               should not print cells which possess borders or shading but which
               extend beyond the last row or column containing data.
paper_type     A value from 0 to 17 corresponding to one of the following paper
               types:
               0    US Letter
               1    Tabloid
               2    Ledger
```

3	Legal
4	Statement
5	Executive
6	Envelope 10
7	Envelope 9
8	Envelope 6
9	A3
10	A4
11	A5
12	B4
13	B5
14	Envelope C4
15	Envelope C5
16	Envelope DIN
17	Custom

See also [SS_SET_PAGE_SETUP@](#)

SS_GET_RANGE_ATTR_INFO@

Returns the text attributes of a range

Format format ss_range_attr_array = SS_GET_ATTR_INFO@(range)

Arguments range A string or array representing a set of cells. A string contains the starting and ending cells in the range. The array contains six elements: leftmost column, top row, rightmost column, bottom row, starting sheet, ending sheet.

Description Returns a formatted array of type [ss_range_attr](#) . The format of this array is defined in the file spsheet_.am.

SS_GET_RANGE_DATA@

Returns the data in a range of cells

Format SS_GET_RANGE_DATA@(range)

Arguments range A string or array representing a set of cells. A string contains the starting and ending cells in the range. The array contains six elements: leftmost column, top row, rightmost column, bottom row, starting sheet, ending sheet.

Description Returns the data of each cell in the specified range. The array returned by the function represents each row of data in the range. For a 2-dimensional range, a 2-dimensional array is returned. The top row is always returned first.

For a 3-dimensional range (which includes more than one sheet), a 3-dimensional array is returned. Rows in the top sheet are returned first, then rows in the second sheet, and so on.

SS_GET_RANGE_PROT@

Indicates whether cells in a range are protected

Format flag = SS_GET_RANGE_PROT@(range[, fullscan])

Method flag = this.get_range_prot@(range[, fullscan])

Arguments

range	The range for which you want protection information. Multiple ranges can be specified.
fullscan	A Boolean value which if set to TRUE indicates that the result of this macro is TRUE only if the entire range is protected.

Description If all the cells in the specified range are protected, SS_GET_RANGE_PROT@ returns TRUE. If one or more of the cells in the specified range are not protected, SS_GET_RANGE_PROT@ returns FALSE.

See also [SS_GET_RANGE_PROT_VIS@](#).

SS_GET_RANGE_PROT_VIS@

Indicates if a range has at least one protected or invisible cell

Format format ss_protection_protState = SS_GET_RANGE_PROT_VIS@(range, fullScan)

Method format ss_protection_protState = this.get_range_prot_vis@(range, fullScan)

Arguments

range	The range for which you want protection and invisibility information. Multiple ranges can be specified.
fullscan	A Boolean value which if set to TRUE indicates that the result of this macro is TRUE only if the entire range is protected or invisible.

Description Returns a format indicating the invisible and protected status of all cells within the specified range. The definition of ss_protection_ is as follows:

format ss_protection_
protected,
invisible,

SS_GET_RANGE_STYLE@

Returns style, precision, and visibility information

Format format ss_style_info = SS_GET_RANGE_STYLE@(range)

Method format ss_style_info = this.get_range_style@(range)

Arguments range The range (as a string) whose number style and precision are of interest.

Description returns a ss_style_format containing information about how data is displayed within the range. The definition of this format is as follows:

format ss_style_

style, The range's number style. Valid number styles include:

- 0 Unstyled
- 1 Boolean
- 2 General
- 3 Fixed
- 4 Scientific
- 5 Currency
- 6 Comma
- 7 Percentage
- 8 Date
- 9 Graph

precision, The range's number precision. Valid precisions are from 0 to 9 for numbers, 0 for alpha characters. If the number style is 8 (date), the precision can be any of the following:

- | | | | |
|---|---------------|----|------------|
| 1 | Mmmm dd, yyyy | 8 | yyyy mm dd |
| 2 | Mmm dd, yyyy | 9 | yy mm dd |
| 3 | dd Mmm yy | 10 | yyyymmdd |
| 4 | mm/dd/yy | 11 | yymmdd |
| 5 | dd.mm.yy | 12 | dd.mm.yyyy |
| 6 | yyyy-mm-dd | 13 | dd/mm/yyy |
| 7 | yy-mm-dd | 15 | Mmmm yyyy |

SS_GET_READ_ONLY@

Indicates whether a file is Read Only

Format flag = SS_GET_READ_ONLY@()

Description Returns TRUE if the current Spreadsheet is a read-only document. Returns FALSE if the current spreadsheet is Read / Write.

SS_GET_ROW_HEIGHT@

Returns the row height of the specified rows

Format height = SS_GET_ROW_HEIGHT@(rowStr, stdHeight)

Method height = this.get_row_height@(rowStr, stdHeight)

Arguments

rowStr	A string indicating the range of rows being examined.
stdHeight	A Boolean value which if set to TRUE indicates that you want information returned indicating if all rows are the standard height.

Description Returns the row height of the specified rows if they all have the same height; otherwise, this macro returns -1.

See also [SS SET ROW HEIGHT@](#)

SS_GET_SHEETS@

Returns the names of all sheets

Format sheetArray = SS_GET_SHEETS@()

Method sheetArray = this.get_sheets@

Description Returns the names of all sheets as an array.

SS_GET_SHEETS_DISPLAY@

Indicates if more than one sheet is displayed

Format flag = SS_GET_SHEETS_DISPLAY@()

Method flag = this.get_sheets_display@

Description Returns TRUE if more than one sheet is displayed. FALSE indicates that only one sheet is displayed.

SS_GET_STATUS@

Returns information on the operating environment

NOTE: This macro is obsolete. Use **SS GET DOC ATTR@** instead.

Format format ss_status_info = SS_GET_STATUS@()

Method format ss_status_info = this.get_status@

Description Returns an array containing the following information. The FORMAT array template name is ss_status_ and is contained in spreadsheet.am.

format ss_status_

open_col,	The number of the column in which the spreadsheet cell cursor is currently positioned. The column number is a zero-based value. Column A has the value 0.
open_row,	The number of the row in which the spreadsheet cell cursor is positioned. The row number is a zero-based value. Row 1 has the value 0.
auto_calc,	Indicates whether the spreadsheet is functioning in automatic or manual calculation mode. 0 manual mode 1 automatic mode
calc_mode,	The spreadsheet's current calculation modes: 0 Natural calculation mode 1 Column calculation mode 2 Row calculation mode
edit_mode,	Indicates whether the spreadsheet is in edit mode. 0 Not in edit mode 1 In edit mode

point_mode,	Reserved. Always returns 0.
bottom_col,	A zero-based column number of the last active cell in the spreadsheet work area. For example, if column G is the last column containing active cells, 6 is returned.
bottom_row,	The zero-based row number of the last active cell in the spreadsheet work area. For example, if the last active cell is in row 5, the value 4 is returned.
auto_graphing,	Reserved. Always returns 1.
calc_count,	Number of calculation iterations when in row or column calculation mode. A number from 1 - 10 is returned. Normal calculation mode returns 1.
font_size,	The typeface size currently set for the spreadsheet. font_size is a number indicating the size. For example, 18 indicates the font's size will be 18 points.
print_size,	Not used.
text_align,	Current text alignment attribute. Possible values are: 0 left 1 right 2 center
num_align,	The current number alignment attribute. Possible values are the same as those for text_align.
num_style,	Current number style attribute. Possible values are: 0 unstyled 1 boolean 2 general 3 fixed 4 scientific 5 currency 6 comma 7 percentage 8 date 9 graph
num_prec,	Current precision setting. num_prec can be 0 to 9 digits. The default is 0. num_prec does not apply to cells having the styles unstyled, date, Boolean, or graph. 15 is returned if the style is set to unstyled.
bold,	Indicates whether the bold typeface attribute is set. 0 not set 1 is set
face,	The current typeface chosen. Possible values are: 0 Times Roman 1 ITC Palatino 2 New Century Schoolbook

	3	Zapf Chancery
	4	Palatino
	5	Courier
	6	Helvetica
	7	Helvetica Narrow
	8	Helvetica
	9	Symbol
	10	Zapf Dingbats
underline,		Indicates whether the underline typeface attribute is set.
	0	not set
	1	is set
italic,		Indicates whether the italic typeface attribute is set.
	0	not set
	1	is set
color,		The color of the text. The values for color can be:
	0	white
	1	black
	2	dark gray
	3	light gray
	4	red
	5	dark red
	6	orange
	7	yellow
	8	olive
	9	green
	10	light blue
	11	blue
	12	dark blue
	13	purple
gridstyle,		Indicates the style in which grids are displayed:
	0	grids are not displayed
	1	dotted grid lines
	2	solid grid lines
wrap_text		A Boolean value which if set to TRUE indicates that text should be wrapped within the cell rather than run off to the right into other cells.

SS_GET_STRUCTURED_COMMENT@

Gets a structured comment from an .as file

Format SS_GET_STRUCTURED_COMMENT@
(commentName)

Method [this.get_structured_comment@](#)
(commentName)

Arguments commentName
A string containing a structured comment within a Spreadsheets document.

Description Gets a structured comment from a Spreadsheets .as file. This comment is invisible from Applixware Spreadsheets, but you can see the structured comment if you use the more command on the file from UNIX. The following shows an example of a structured comment:

```
*BEGIN SPREADSHEETS VERSION=420/420 ENCODING=7BIT  
** "CommentName" CommentValue
```

See also [SS_SET_STRUCTURED_COMMENT@](#)

SS_GET_VIEWS_IN_DOC@

Returns an array of View names
in a Spreadsheets document

Format SS_GET_VIEWS_IN_DOC@(filename)

Method [this.get_views_in_doc@\(filename\)](#)

Arguments filename The path name of an Applixware Spreadsheets file

Description Returns an array containing the views in the specified Spreadsheets file.

See also [SS_GET_VIEW_INFO@](#)

SS_GET_VIEW_FORMULAS@

Return TRUE if formulas are displayed

Format SS_GET_VIEW_FORMULAS@()

Method [this.get_view_formulas@\(\)](#)

Description If formulas are being displayed in the current spreadsheet, this macro returns TRUE (-1). If values are being displayed, this macro returns FALSE (0).

See also [SS_SET_VIEW_FORMULAS@](#)

SS_GET_VIEW_INFO@

Return information about a named view

Format SS_GET_VIEW_INFO@(viewName)

Method [this.get_view_info@\(viewName\)](#)

Arguments viewName A named view in the current spreadsheet

Description Returns an array that contains information about a named view. The array is of format SS_VIEW_INFO@. The SS_VIEW_INFO@ format is structured as follows:

format ss_view_info@

sheet,	'0 based sheet number
default_width,	'default column width in chars
default_height,	'default height of row in points
title_rows,	'array of title rows in this view
title_cols,	'array of title columns in this view
format arrayof ss_row_pair@ row_heights,	'array of rows that are not set to ' the default row height, and the ' height of those rows.
format arrayof ss_col_pair@ col_widths,	'array of columns that are not set ' to the default column width, and ' the width of those columns
format arrayof ss_view_pair@ visible_rows,	

This format contains an array containing the starting and ending row designations of each visible block of rows in the view. For example, if row 3 is invisible, two row pairs are returned: 0 and 1 (starting and ending row designation for the top visible block), and

3 and 32766 (starting and ending row designation for the bottom visible block.) These are zero-based values.

format arrayof ss_view_pair@ visible_cols

This format contains an array containing the starting and ending column designations of each visible block of columns in the view. For example, if column C is invisible, two column pairs are returned: 0 and 1 (starting and ending columns for the leftmost visible block), and 3 and 701 (starting and ending column designation for the rightmost visible block). These are zero-based values.

format ss_col_pair@

col, '0-based column number
width 'width in characters

format ss_view_pair@

first, ' start of visible row or column block
last ' end of visible row or column block

format ss_row_pair@

row, '0-based row number
height 'height of the row in points

SS_GET_WORKSHEET_NAME@

Returns the name of the specified worksheet

Format name = SS_GET_WORKSHEET_NAME@(sheetNumber)

Method name = this.get_worksheet_name@(sheetNumber)

Arguments sheetNumber The number of the worksheet whose name you want to retrieve. This number is zero-based.

Description Returns the name of a worksheet. The sheetNumber is zero-based. When you open an empty spreadsheet, these numbers and names are established by default:

0 is A

1 is B

2 is C

3 is D

See also [SS_GET_SHEETS@](#)

SS_SET_WORKSHEET_NAME@

SS_GET_ZOOM_FACTOR@

Returns the current zoom factor

Format zoomFactor = SS_GET_ZOOM_FACTOR@()

Method zoomFactor = this.get_zoom_factor@

Description Returns the current zoom factor as an integer value.

SS_GOAL_SEEK@

Determines a number that satisfies a given formula based on the desired final result for the formula

Format SS_GOAL_SEEK@(formula, result, value[, precision])

Method this.goal_seek@(formula, result, value[, precision])

Arguments

formula	A string indicating the cell which contains a formula for which you have set a target value. For example, "B6" indicates that cell B6 contains the target formula.
result	A string indicating the cell in which the goal value will be placed.
value	The desired value for the formula contained in formula. value must be a number.
precision	An optional number indicating the number of digits of precision to which the goal value should be calculated. If precision is not specified, .01 is used.

Description Determines the value necessary to obtain a desired result for a formula. The value necessary to achieve the result is placed in the cell specified as result. For example, suppose the formula in formula is +A1+1 and the desired value for the formula is 20. If you specify "A1" as the result, SS_GOAL_SEEK@ will place the value 19 in cell A1, since 19+1 = 20.

If the specified goal cannot be reached, an error is thrown.

SS_GOTO_BEGIN_LINE@

Moves to the beginning of the line

Format SS_GOTO_BEGIN_LINE@()

Method [this.goto_begin_line@](#)

Description Moves the cursor before the first character in the entry area after the justification character. The spreadsheet must be in edit mode. For example, the " ^ " indicates the position in a line to which the cursor is moved if the current text is right justified:

" ^ Annual Income

SS_GOTO_BEGIN_LINE@ is called by the Keys ® Beginning of line menu option.

See also [SS_GOTO_EOL@](#)

SS_GOTO_EOL@

Moves the cursor to the end of the entry line area

Format SS_GOTO_EOL@()

Method [this.goto_eol@](#)

Description The cursor is placed after the last character in the entry area in the current Spreadsheets document. The spreadsheet must be in edit mode. SS_GOTO_EOL@ is called by Keys ® End of line.

See also [SS_GOTO_BEGIN_LINE@](#)

SS_GO_BEGIN@

Moves the cursor to cell A1

Format SS_GO_BEGIN@()

Method [this.go_begin@](#)

Description Moves the cursor to cell A1. This macro is called by the Find ® Home menu option.

See also [SS_GO_END@](#)

SS_GO_CELL@

SS_GO_CELL@

Moves the cursor to the specified cell

Format SS_GO_CELL@(cell)

Method [this.go_cell@](#)(cell)

Arguments cell A string indicating the cell to which to move the cursor. cell can be an individual cell address, such as "B9," or it can be a range of cells. A range of cells can be specified using a range name or the range cell addresses, such as "B3..E5".

Description Moves the cursor to the specified cell location. If cell is a range, SS_GO_CELL@ moves the cursor to the top left cell of the range.

See also [SS_GO_BEGIN@](#)

[SS_GO_END@](#)

SS_GO_END@

Moves the cursor to the *end* of the data

Format SS_GO_END@()

Method [this.go_end@](#)

Description Moves the cursor to the bottom right cell of the last column and row containing data. SS_GO_END@ is called by the Find ® End menu option.

See also [SS_GO_BEGIN@](#)

[SS_GO_CELL@](#)

SS_GO_NAME_RANGE@

Selects and moves to a range

Format SS_GO_NAME_RANGE@(range)

Method [this.go_name_range@\(range\)](#)

Arguments range The range (as a string) you want to select.

Description Selects a specified range and moves the cursor to the bottom right cell of the range. [SS_GO_NAME_RANGE@](#) is called by the Find ® Named Range menu option.

SS_GO_NEXT_SELECTION@

Goes to the next selection

Format [SS_GO_NEXT_SELECTION@\(\)](#)

Method [this.go_next_selection@](#)

Description Goes to the next selection and repaints the screen and edit line at that position.

See also [SS_GO_PREVIOUS_SELECTION@](#)

SS_GO_PREVIOUS_SELECTION@

Goes to the previous selection

Format [SS_GO_PREVIOUS_SELECTION@\(\)](#)

Method [this.go_previous_selection@](#)

Description Goes to the previous selection and repaints the screen and edit line at that position.

See also [SS_GO_NEXT_SELECTION@](#)

SS_GRID_LINES@

Toggles the display of grid lines

Format [SS_GRID_LINES@\(\)](#)

Method [this.grid_lines@](#)

Description If grid lines are not currently displayed in the spreadsheet, [SS_GRID_LINES@](#) displays grid lines. If grid lines are displayed, [SS_GRID_LINES@](#) turns off the display.

Grid lines are displayed as dotted lines or solid lines, depending on the preferences setting. `SS_GRID_LINES@` is called by the View ® Grid lines menu option.

See also [SS_GRID_LINES_ON@](#)
[SS_GRID_LINES_OFF@](#)
[SS_SET_GRID_LINES@](#)

SS_GRID_LINES_OFF@

Disables the display of grid lines

Format `SS_GRID_LINES_OFF@()`

Method `this.grid_lines_off@`

Description Forces grid lines to disappear from a Spreadsheets document.

See also [SS_GRID_LINES_ON@](#)
[SS_GRID_LINES@](#)
[SS_SET_GRID_LINES@](#)

SS_GRID_LINES_ON@

Enables the display of grid lines

Format `SS_GRID_LINES_ON@()`

Method `this.grid_lines_on@`

Description Forces grid lines to appear in a Spreadsheets document.

See also [SS_GRID_LINES_OFF@](#)
[SS_GRID_LINES@](#)
[SS_SET_GRID_LINES@](#)

SS_IMPORT_XLS@

Imports a XLS file into a new Spreadsheets window

Format SS_IMPORT_XLS@(file, revertFlag)

Method [this.import_xls@](#)(file, revertFlag)

Arguments

file	The full path name, a string, of the XLS file to import.
revertFlag	A Boolean value which if set to TRUE indicates that the data will be imported into the current Spreadsheets window. FALSE indicates it will be imported into a new window.

Description Imports an XLS file into a new Spreadsheets window. Material imported from the XLS file may not be the same as it appeared in the application from which the file originated. Any functions or formulas not supported by Spreadsheets are displayed as text strings. A list of conversion errors resulting from the import will appear in a box entitled **[Import Problems](#)**.

SS_INSERT@

Inserts rows or column

Format SS_INSERT@(insertPoint, amount)

Method [this.insert@](#)(insertPoint, amount)

Arguments

insertPoint	A string indicating the column or row where the columns or rows should be inserted. Use column letter for columns and row numbers for rows. Rows are numbered from 1. Information is inserted before insertPoint. For example, insertPoint "c" indicates that columns will be inserted before column C.
amount	A number indicating how many rows or columns to insert.

Description Inserts rows or columns. When inserting columns, data from the insertion point shifts to the right. For example, if two columns are inserted before column C, the data from column C is shifted to column E.

When inserting rows, data beginning at the insertion point shifts down. For example, if five rows are inserted before row 8, the data from row 8 is shifted to row 13.

If data is shifted, cell references to this data are automatically updated to reflect the new cell location.

See also [SS_DELETE@](#)
[SS_INSERT_COLS@](#)
[SS_INSERT_ROWS@](#)

SS_INSERT_BUTTON@

Inserts a button in the Spreadsheet at the cursor position

Format SS_INSERT_BUTTON@(name, title, execmacro, bitmap_path, aslink)

Arguments

name	A string. The name of the button object within the Spreadsheet.
title	A string. This string name appears on the button, unless you specify a bitmap.
execmacro	A string. The macro that runs when you click the button in the Spreadsheet.
bitmap_path	A string. The path of an Applixware bitmap file.
aslink	A Boolean. If TRUE, the bitmap displayed on the button is a link to an external file. If FALSE, the bitmap displayed on the button is embedded in the Spreadsheet document.

Description Inserts a button in the Spreadsheets document at the cursor location.

SS_INSERT_CLIPART@

Inserts any graphic object into the spreadsheet

Format SS_INSERT_CLIPART@(gfx)

Method [this.insert_clipart@\(gfx\)](#)

Arguments gfx A graphics handle.

Description Inserts the graphic objects whose graphic handle is gfx into the spreadsheet.

This macro invokes [SS_DRAG_INSET_OBJECT@](#) which allows the user to place the object in the spreadsheet and drag the mouse to indicate the size at which the graphic object will be displayed.

If there is a need to manipulate the contents of the graphic object, the object must be manipulated using its graphic handle and the GFX series of Graphic ELF macros.

SS_INSERT_COLS@

Inserts columns

Format SS_INSERT_COLS@(insertPoint, amount, whereFlag)

Method [this.insert_cols@](#)(insertPoint, amount, whereFlag)

Arguments

insertPoint	A string indicating the column where the columns should be inserted. Use column letter for columns. insertPoint can be: <ul style="list-style-type: none">· A single column.· A column on one or more sheets specified in the form beginning sheet..end sheet.
amount	A number indicating how many columns to insert.
whereFlag	If TRUE, columns are inserted left of the cursor location. If FALSE, columns are inserted RIGHT of the cursor location.

Description Inserts blank columns.

Information is inserted either before or after insertPoint, depending on the whereFlag setting. For example, if whereFlag = TRUE and insertPoint = "c", columns will be inserted before column C on the first sheet. If insertPoint = "A:C..B:C", columns will be inserted before column C on sheets A through B.

When inserting columns, data from the insertion point shifts to the right. For example, if two columns are inserted before column C, the data from column C is shifted to column E.

If data is shifted, cell references to this data are automatically updated to reflect the new cell location.

See also [SS_DELETE@](#)
[SS_DELETE_COLS@](#)
[SS_INSERT@](#)

SS_INSERT_LINE_BREAK@

Inserts a line break into the edit line

Format SS_INSERT_LINE_BREAK@()

Method [this.insert_line_break@](#)

SS_INSERT_OBJECT_AT_CELL@

Inserts the graphic object in a cell

Format SS_INSERT_OBJECT_AT_CELL@(name, cellStr, width, height, xULoff, yULoff, xLRoff, yLRoff)

Method [this.insert_object_at_cell@\(name, cellStr, width, height, xULoff, yULoff, xLRoff, yLRoff\)](#)

Arguments	name	The name of the object being inserted.
	cellStr	The cell into which the object is being inserted.
	width	The object's width.
	height	The object's height.
	xULoff	The object's upper left X offset position.
	yULoff	The object's upper left Y offset position.
	xLRoff	The object's lower right X offset position.
	yLRoff	The object's lower right Y offset position.

Description Inserts and draws a graphic object at a cell and at a location within the cell. The height and width of the cell is always considered to be 1 unit. The offset positions of the inserted object are indicated by fractional amounts of this unit. For example, assume that the cell is 2 inches by 3 inches and the upper left corner may be the relative coordinate (.2, .25). Thus, the object's upper left coordinate will be drawn .4 inches to the right and .5 inches down from the cell's upper left corner.

SS_INSERT_ROWS@

Inserts rows or column

Format SS_INSERT_ROWS@(insertPoint, amount, whereFlag)

Method `this.insert_rows@(insertPoint, amount, whereFlag)`

Arguments

<code>insertPoint</code>	A string indicating the row where the rows should be inserted. Use row numbers for rows. Rows are numbered from 1. <code>insertPoint</code> can be: <ul style="list-style-type: none">· A single row.· A row on one or more sheets specified in the form beginning sheet..end sheet.
<code>amount</code>	A number indicating how many rows or columns to insert.
<code>whereFlag</code>	A Boolean. If TRUE, the rows are inserted before the cursor location. If FALSE, rows are inserted after the cursor location.

Description Inserts rows. When inserting rows, data beginning at the insertion point shifts down. For example, if five rows are inserted before row 8, the data from row 8 is shifted to row 13.

Information is inserted either before or after `insertPoint` depending on the `whereFlag`. For example:

`amount = 2`

`whereFlag = TRUE`

`insertPoint = 2`

`SS_INSERT_ROWS@(insertPoint, amount, whereFlag)`

Two rows are inserted before row 2.

`amount = 2`

`whereFlag = TRUE`

`insertPoint = "A:2..B:2"`

`SS_INSERT_ROWS@(insertPoint, amount, whereFlag)`

Two columns will be inserted before row 2 on sheets A through B.

If data is shifted, cell references to this data are automatically updated to reflect the new cell location.

See also [**SS_DELETE@**](#)
[**SS_DELETE_ROWS@**](#)
[**SS_INSERT@**](#)

SS_INSERT_SHEETS@

Creates a new sheet

Format label = SS_INSERT_SHEETS@(insertionPont, count)

Method label = this.insert_sheets@(insertionPont, count)

Arguments insertPoint A string indicating where the sheets should be inserted. New sheets will be inserted before the indicated sheet letter.
count A number indicating how many sheets to insert.

Description Inserts a new sheet into the spreadsheet. If you insert an entire sheet, the subsequent sheets move backward. For example, if you insert a sheet before sheet C, sheet C becomes sheet D, sheet D becomes sheet E and so on.

SS_INSTALL_ADDIN_FUNCTIONS@

Loads the add-in functions contained in libss.so

Format SS_INSTALL_ADDIN_FUNCTIONS@()

Method this.install_addin_functions@

Description Loads and installs all of the shared library functions contained in the libss.so shared library.

SS_INVISIBLE@

Makes cells invisible

Format SS_INVISIBLE@(range)

Method this.invisible@(range)

Arguments range The range of cells that should be made invisible. Multiple ranges can be specified.

Description Makes the contents of cells that are visible invisible. The contents of invisible cells do not appear in the spreadsheet, although an invisible cell's contents will be displayed in

the entry area when the cursor is in the cell. Invisible cells appear empty when printed. If an invisible cell is edited, it becomes visible.

See also [SS_INVISIBLE_SELECTED@](#)
[SS_VISIBLE@](#).

SS_INVISIBLE_SELECTED@

Makes selected cells invisible

Format SS_INVISIBLE_SELECTED@()

Method [this.invisible_selected@](#)

See also [SS_INVISIBLE@](#)
[SS_VISIBLE@](#).

SS_IS_ERROR@

Determines whether an elf datum is an ERROR status

Format SS_IS_ERROR@(elfdatum)

Arguments elfdatum

Description Tests an ELF datum to determine whether it is an ERROR status. If the datum is an ERROR status, the macro returns TRUE (-1). If the datum is not an ERROR status, the macro returns FALSE (0). This macro is the ELF macro equivalent of the built-in spreadsheets function ERROR().

See also [SS_MAKE_ERROR_DATA@](#)

Example

SS_IS_NA@

Determines whether an elf datum is an NA status

Format SS_IS_NA@(elfdatum)

Method [IS_NA@\(elfdatum\)](#)

Arguments elfdatum

Description Tests an ELF datum to determine whether it is an NA status. If the datum is an NA status, the macro returns TRUE (-1). If the datum is not an NA status, the macro returns FALSE (0). This macro is the ELF macro equivalent of the built-in spreadsheets function NA().

See also [SS_MAKE_NA_DATA@](#)

SS_IS_SHEET_LOCKED@

Returns TRUE if the sheet is locked

Format flag = SS_IS_SHEET_LOCKED@(sheet)

Arguments sheet A sheet letter

Description Returns TRUE if the specified sheet is locked. A locked sheet cannot be altered in any way until it is unlocked. You can cut and paste the values in a locked sheet to another sheet, however.

See also [SS_LOCK_SHEET@](#)
[SS_UNLOCK_SHEET@](#)

SS_ITALICS@

Makes text or cell italic

Format SS_ITALICS@()

Method [this.italics@](#)

Description Makes the selected text italic. If the selection is already italicized, SS_ITALICS@ removes the italic attribute from the selection.

If no text is selected, the current cell is italicized.

SS_ITALICS@ is called by the Style ® Italic menu option.

See also [SS_BOLD@](#)
[SS_UNDERLINE@](#)

SS_HELP_KEY@

Invokes Help On Context

Format SS_HELP_KEY@()

Method [this.help_key@](#)

SS_HELP_MODE@

Invokes help on context mode

Format SS_HELP_MODE@()

Method [this.help_mode@](#)

SS_HIDE@

Hides rows and/or columns

Format SS_HIDE@(list)

Method [this.hide@\(list\)](#)

Arguments list A string indicating the rows and columns to hide. Each row number or column letter in list is separated by a comma. For example, to hide columns E and F and rows 6 and 8, you specify list as "E,F,6,8".

Description Hides the specified rows or columns from display. Hidden rows and columns can be displayed by calling **[SS REVEAL SOME@](#)**.
Cells are only hidden on the named or current sheets.

SS_IMPORT_ASCII@

Imports an ASCII file to Applixware Spreadsheets

Format SS_IMPORT_ASCII@(filename, ascType, revertFlag)

Method [this.import_ascii@\(filename, ascType, revertFlag\)](#)

Arguments	filename	The name of the ASCII file being imported into Applixware Spreadsheets.
	ascType	One of the following values: <ul style="list-style-type: none"> 8 Imports the named DIF/XDIF file. See SS_IMPORT_DIF@. 9 Imports data, placing information in cells based on the delimiters within file name. See SS_IMPORT_GRID@. 10 Imports column by column. See SS_IMPORT_COL@. 11 Imports row by row. See SS_IMPORT_ROW@. 12 Input file contains comma separated values. See SS_IMPORT_CSV@.
	revertFlag	A Boolean value which if set to TRUE indicates that the data will be imported into the current Spreadsheets window. FALSE indicates it will be imported into a new window.

Description Imports ASCII data into a spreadsheet. This macro is called by the five macros indicated in the ascType parameter. See these macros for a description of the action performed.

SS_IMPORT_COL@

Imports an ASCII file into a Applixware Spreadsheets window placing each line into a different cell, column by column

Format SS_IMPORT_COL@(file, revertFlag)

Method [this.import_col@](#)(file, revertFlag)

Arguments	file	The full path name, a string, of the ASCII file to import.
	revertFlag	A Boolean value which if set to TRUE indicates that the data will be imported into the current Spreadsheets window. FALSE indicates it will be imported into a new window.

Description Imports an ASCII file into a new Spreadsheets window. Each line of the ASCII file is placed in a separate contiguous cell in the spreadsheet, starting with cell A1 (A1, B1, C1, D1, and so on). See also [SS_PASTE_IMPORT_CURDOC@](#).

See also [SS_IMPORT_ASCII@](#)

SS_IMPORT_CSV@

Imports a Comma Separated Values file

Format SS_IMPORT_CSV@(file, revertFlag)

Method [this.import_csv@\(file, revertFlag\)](#)

Arguments

file	The full path name, a string, of the CSV file to import.
revertFlag	A Boolean value which if set to TRUE (-1) indicates that the data will be imported into the current Spreadsheets window. FALSE (0) indicates it will be imported into a new window.

Description Imports a "Comma Separated Values" (CSV) file into a new Spreadsheets window. Material imported from the CSV file may not be the same as it appeared in the application from which the file originated. Any functions or formulas not supported by Spreadsheets are displayed as text strings.

A list of conversion errors resulting from the import will appear in a box entitled **[Import Problems](#)**.

See also [SS_IMPORT_ASCII@](#)

SS_IMPORT_DIF@

Imports a DIF/XDIF file into a new Spreadsheets window

Format SS_IMPORT_DIF@(file, revertFlag)

Method [this.import_dif@\(file, revertFlag\)](#)

Arguments

file	The full path name, a string, of the DIF file to import.
revertFlag	A Boolean value which if set to TRUE (-1) indicates that the data will be imported into the current Spreadsheets window. FALSE (0) indicates it will be imported into a new window.

Description Imports a DIF/XDIF file into a new Spreadsheets window. Material imported from the DIF/XDIF file may not be the same as it appeared in the application from which the file originated. Any functions or formulas not supported by Spreadsheets are displayed as text strings.

A list of conversion errors resulting from the import will appear in a box entitled **[Import Problems](#)**.

See also [SS_IMPORT_ASCII@](#)

SS_IMPORT_GRID@

Imports an ASCII file into a new Spreadsheets grid

Format SS_IMPORT_GRID@ (file, revertFlag)

Method [this.import_grid@](#) (file, revertFlag)

Arguments

file	The full path name, a string, of the ASCII file to import.
revertFlag	A Boolean value which if set to TRUE (-1) indicates that the data will be imported into the current Spreadsheets window. FALSE (0) indicates it will be imported into a new window.

Description Imports the specified ASCII file into a new Spreadsheets window. The information in the ASCII file is placed in separate cells in the spreadsheet, based on the location of delimiters in the ASCII file. The delimiter(s) are determined by the ASCII Delimiter setting in the * ® Spreadsheets Preferences menu option. All information between delimiters is placed in a single cell.

See also [SS_IMPORT_ASCII@](#)

SS_IMPORT_ROW@

Imports an ASCII file into a Applixware Spreadsheets window, placing each line into a different cell, row by row

Format SS_IMPORT_ROW@(file, revertFlag)

Method [this.import_row@](#)(file, revertFlag)

Arguments

file	The full path name, a string, of the ASCII file to import.
revertFlag	A Boolean value which if set to TRUE indicates that the data will be imported into the current Spreadsheets window. FALSE indicates it will be imported into a new window.

Description Imports an ASCII file into a new Spreadsheets window. Each line of the ASCII file is placed in a separate contiguous cell in the spreadsheet, starting with cell A1 (A2, A3, A4, and so on).

See also [SS_IMPORT_ASCII@](#)

SS_PASTE_IMPORT_CURDOC@

SS_IMPORT_SYLK@

Imports a SYLK file into a new Spreadsheets window

Format SS_IMPORT_SYLK@(file, revertFlag)

Method [this.import_sylk@](#)(file, revertFlag)

Arguments file The full path name, a string, of the SYLK file to import.
revertFlag This must be set to FALSE.

Description Imports a SYLK file into a new Spreadsheets window. Material imported from the SYLK file may not be the same as it appeared in the application from which the file originated. Any functions or formulas not supported by Spreadsheets are displayed as text strings. A list of conversion errors resulting from the import will appear in a box entitled **[Import Problems](#)**.

SS_IMPORT_WK3@

Imports a WK3 file into a new Spreadsheets window

Format SS_IMPORT_WK3@(file, revertFlag)

Method [this.import_wk3@](#)(file, revertFlag)

Arguments file The full path name, a string, of the WK3 file to import.
revertFlag A Boolean value which if set to TRUE indicates that the data will be imported into the current Spreadsheets window. FALSE indicates it will be imported into a new window.

Description Imports a WK3 file into a new Spreadsheets window. Material imported from the WK3 file may not be the same as it appeared in the application from which the file originated. Any functions or formulas not supported by Spreadsheets are displayed as text strings. A list of conversion errors resulting from the import will appear in a box entitled **[Import Problems](#)**.

SS_IMPORT_WKS@

Imports a WK1 file into a new Spreadsheets window

Format SS_IMPORT_WKS@(file, revertFlag)

Method [this.import_wks@](#)(file, revertFlag)

Arguments

file	The full path name, a string, of the WKS file to import.
revertFlag	A Boolean value which if set to TRUE indicates that the data will be imported into the current Spreadsheets window. FALSE indicates it will be imported into a new window.

Description Imports a WKS file into a new Spreadsheets window. Material imported from the WKS file may not be the same as it appeared in the application from which the file originated. Any functions or formulas not supported by Spreadsheets are displayed as text strings. A list of conversion errors resulting from the import will appear in a box entitled **Import Problems**.

SS_JUSTIFY_?@

Sets the justification for the cells specified

Format SS_JUSTIFY_?@()

Method [this.justify_?@](#)

Description Sets the justification attribute for the cells you specify. Justification is applied to empty cells and cells containing values. Here are the justification macros.

SS_JUSTIFY_DEFAULT@()

Sets the justification of the current selection to the default settings as defined via Style ® Document Settings.

SS_JUSTIFY_LEFT@()

Left-justifies all cells in the current selection.

SS_JUSTIFY_RIGHT@()

Right-justifies all cells in the current selection.

SS_JUSTIFY_CENTER@()

Centers all cells in the current selection.

SS_JUSTIFY_REPEAT@()

For each cell in the selection, causes the string of characters to be repeated until the cell is filled.

See also [SS_CELL_JUSTIFY@](#)
[SS_JUSTIFY_SELECTED@](#)

SS_JUSTIFY_SELECTED@

Sets the justification for a selection

Format SS_JUSTIFY_SELECTED@(type)

Method [this.justify_selected@](#)(type)

Arguments type One of the following justification settings:

- 0 no justification
- SSC#JUST_LEFT left justification
- SSC#JUST_RIGHT right justification
- SSC#JUST_CENTER center justification
- SSC#JUST_REPEAT repeat

Description Sets the justification for the current selection to type. If the number you type is less than 0 or greater than 5, Spreadsheets uses the default justification.

See also [SS_CELL_JUSTIFY@](#)
[SS_JUSTIFY ?@](#)

SS_LARGER_KEY@

Makes the column 1 character wider

Format SS_LARGER_KEY@()

Method [this.larger_key@](#)

See also [SS_SMALLER_KEY@](#)

SS_LAST_CELL_INFO@

Returns information describing a cell's previous state

Format format ss_cell_info_info = SS_LAST_CELL_INFO@()

Method format ss_cell_info_info = this.last_cell_info@

Description Returns ss_cell_info_formatted information that describes the state of the cell before it was changed. The definition of this format is as follows:

```
format ss_cell_info_
  col,          'zero-based column number, as passed
  row,          'zero-based row number, as passed
  type,        'cell types
                100 SSC#CELL_NUM_FORMULA
                101 SSC#CELL_TEXT_FORMULA
                102 SSC#CELL_BOOL_FORMULA
                103 SSC#CELL_OBSOLETE
                104 SSC#CELL_ERROR
                105 SSC#CELL_NA
                106 SSC#CELL_PENDING
                107 SSC#CELL_CIRCULAR
                108 SSC#CELL_DIVZERO
                109 SSC#CELL_UDEFNAME
                110 SSC#CELL_NUMERR
                111 SSC#CELL_TYPERR
                112 SSC#CELL_REFERR
                113 SSC#CELL_ARGERR
                114 SSC#CELL_FAILERR
                115 SSC#CELL_DBERR1
                116 SSC#CELL_DBERR2
                117 SSC#CELL_DBERR3
                200 SSC#CELL_NUMERIC
                201 SSC#CELL_TEXT
                204 SSC#CELL_EMPTY
                205 SSC#CELL_NA_VAL
                206 SSC#CELL_ERROR_VAL
                2   SSC#CELL_IS_OBSOLETE_
                3   SSC#CELL_IS_ERROR_
                4   SSC#CELL_IS_NA_
                5   SSC#CELL_IS_PENDING_
                6   SSC#CELL_IS_CIRUCULAR_
```

display_str,	'grid display string
entry_str,	'entry line string
protected,	'Boolean
style_type,	'display style type
precision,	'extent of display style
value,	'current value if cell is numeric
sheet	

SS_LEFT_ARROW_KEY@

Moves the cell pointer one cell to the left

Format SS_LEFT_ARROW_KEY@()

Method [this.left_arrow_key@](#)

See also [SS_BACK_RETURN_KEY@](#)
[SS_DOWN_ARROW_KEY@](#)
[SS_RETURN_KEY@](#)
[SS_RIGHT_ARROW_KEY@](#)
[SS_UP_ARROW_KEY@](#)

SS_LEFT_SCREEN_KEY@

Displays information to the left of the current display

Format SS_LEFT_SCREEN_KEY@()

Method [this.left_screen_key@](#)

See also [SS_RIGHT_SCREEN_KEY@](#)

SS_LEFT_SECTION@

Moves the cursor left to the next cell containing data

Format SS_LEFT_SECTION@()

Method [this.left_section@](#)

Description Moves the cursor from its current position to the next cell on the left that contains data. If no cells containing data are found to the left of the current cursor position, the cursor is moved to the first cell in the row.

SS_LEFT_SECTION@ is called by the Keys ® Next data left menu option.

See also [SS_RIGHT_SECTION@](#)

SS_LINK_TO_INSET@

Links a file inset into a Spreadsheets document

Format SS_LINK_TO_INSET@(pathname, docType, execMacro, filterMacro)

Arguments

pathname	The absolute pathname of a file to embed at the cursor location in the current Spreadsheet.
docType	An integer indicating the type of document. The valid doctypes are listed in the file recgfil_.am.
execMacro	The macro executed when you double-click the inset.
filterMacro	The filter macro converts a foreign file into Applixware format to display it in an Applixware document. You can enter an alternative file filter program you have created to convert the file instead of the Applixware-supplied filter. Leave this option blank to use the default conversion macro.

Description The file specified by pathname displays in the Spreadsheet at the cursor location, but exists in the file system as a file separate from the .as file.

See also [SS_EMBED_INSET@](#)

SS_LOAD_FILE@

Loads a file

Format SS_LOAD_FILE@(file, readOnlyFlag)

Method [this.load_file@](#)(file, readOnlyFlag)

Arguments

file	The name of the spreadsheet file you want to load. If the file is not in the current directory, file must be a full path name.
------	--

readOnlyFlag A boolean. If this is set to TRUE, the file is loaded into the spreadsheet in read-only mode. If this is set to FALSE, the file is loaded in read/write mode.

Description Loads the specified file into the current spreadsheet. If you have not saved changes to the current spreadsheet, the changes are lost. If the document specified by name does not exist, a blank spreadsheet is displayed in the current window and the window is given the name specified by name.

SS_LOAD_VIEW@

Loads the specified view

Format SS_LOAD_VIEW@(view)

Method [this.load_view@\(view\)](#)

Arguments view The name of an existing view.

Description Loads the specified view and makes it the current view. If the specified view does not exist, an error is thrown.

SS_LOCALIZE_EXTERNAL_LINKS@

Makes an external link local

Format SS_LOCALIZE_EXTERNAL_LINKS@(rangename)

Method [this.localize_external_links@\(rangename\)](#)

Arguments rangename A range containing a reference to an external link.

Description Changes the link pointed to by rangename from an external link to a local link.

SS_LOCALIZE_LINKS@

Localizes links or objects

Format SS_LOCALIZE_LINKS@(name, type)

Method [this.localize_links@\(name, type\)](#)

Arguments	name	The name of a range or the name of an object
	type	The type of localization that will occur:
	1	Named Range localization.
	2	Object localization.

Description Localizes the objects pointed to by name. That is, the object is brought within the current spreadsheet.

SS_LOCALIZE_OBJECT_LINKS@

Makes the current object a local object

Format SS_LOCALIZE_OBJECT_LINKS@(name)

Method [this.localize_object_links@\(name\)](#)

Arguments	name	The name of the object being localized.
------------------	------	---

SS_LOCK_SHEET@

Locks a sheet in a Spreadsheets document

Format SS_LOCK_SHEET@(sheet, passwd)

Arguments	sheet	The sheet letter. This sheet will be locked.
	passwd	A string. This password is used in unlocking the sheet. The password is case sensitive. It is stored in plain text in the Spreadsheets document when the file is saved.

See also [SS_UNLOCK_SHEET@](#)

SS_MAIL@

Mails the current Spreadsheet document

Format SS_MAIL@()

Method [this.mail@](#)

Description Sends the current Spreadsheets document to someone using Applixware Mail.

SS_MAKE_ERROR_DATA@

Creates an ERROR datum in an ELF Macro

Format SS_MAKE_ERROR_DATA@()

Method [this.make_error_data@\(\)](#)

Description Creates an ERROR datum in an ELF macro. This macro is equivalent to the built-in function ERROR() in Spreadsheets.

See also [SS IS ERROR@](#)

SS_MAKE_NA_DATA@

Creates an NA datum in an ELF macro

Format SS_MAKE_NA_DATA@()

Method [this.make_na_data@\(\)](#)

Description Creates an NA datum in an ELF macro. This macro is equivalent to the built-in macro NA() in Spreadsheets.

See also [SS IS NA@](#)

SS_MODIFIED@

Indicates whether a spreadsheet is modified

Format flag = SS_MODIFIED@()

Method flag = this.modified@

Description Returns TRUE if the current spreadsheet has been modified since it was last saved to a file. Returns FALSE if no modifications were made.

If the spreadsheet is new and has never been saved to a file, TRUE is returned if any editing occurred in the spreadsheet.

SS_MOVE_CELLS_DOWN@

Inserts cells down

Format SS_MOVE_CELLS_DOWN@(insertPoint, count)

Method this.move_cells_down@(insertPoint, count)

Method MAKE_NA_DATA@()

Arguments insertPoint The cell at which insertion begins.

count The number of cells being inserted.

Description Inserts count cells into the spreadsheet beginning at cell insertPoint. Cells are inserted in the *down* order. That is, the cells are inserted in the same column beginning at cell insertPoint. As the cells are inserted, the current contents of those cells are shifted down.

SS_MOVE_CELLS_LEFT@

Moves row to the left

Format SS_MOVE_CELLS_LEFT@(cell, count)

Method this.move_cells_left@(cell, count)

Arguments cell The first cell into which cells to the left are moved.

count The number of cells to the left of cell.

Description Moves cells to the left. All cells in the row are moved left beginning at the cell that is count cells to the right of cell. In other words, count cells are deleted beginning at position cell and the remainder of the row is moved. For example:

SS_MOVE_CELLS_LEFT@("C20", 9)

This macro moves the contents of L20 into C20, M20 into D20, and so on.

SS_MOVE_CELLS_RIGHT@

Inserts cells right

Format SS_MOVE_CELLS_RIGHT@(insertPoint, count)

Method [this.move_cells_right@\(insertPoint, count\)](#)

Arguments insertPoint The cell at which insertion begins.

count The number of cells being inserted.

Description Inserts count cells into the spreadsheet beginning at cell insertPoint. Empty cells are inserted. As cells are being inserted, the data that existed previously in these is moved (shifted) right.

SS_MOVE_CELLS_UP@

Moves column up

Format SS_MOVE_CELLS_UP@(cell, count)

Method [this.move_cells_up@\(cell, count\)](#)

Arguments cell The first cell into which cells are moved.

count The number of cells from cell at which the movement occurs.

Description Moves cells up. All cells in the column are moved up beginning at the cell that is count cells below cell. In other words, count cells are deleted beginning at position cell and the remainder of the column is moved. For example:

SS_MOVE_CELLS_UP@("C20", 9)

This macro moves the contents of C29 into C20, C30 into C21, and so on.

SS_NAME_CHANGE@

Changes a range name

Format SS_NAME_CHANGE@(oldName, newName, range)

Method [this.name_change@\(oldName, newName, range\)](#)

Arguments

oldName	A string indicating the name by which the range is currently identified.
newName	A string indicating the new name by which the specified range will be identified.
range	A string indicating the range to which the name is assigned.

Description Changes a range name or assigns a different range to an existing range name. You can use SS_NAME_CHANGE@ to change the name of a range by specifying a new range name with the range argument. Range names should not include the following spreadsheet reserved characters or reserved words: + * ^ / ,) (> < = # . \$ ERROR, ERR, TRUE, FALSE, RAND, TODAY, and PI.

You can also use SS_NAME_CHANGE@ to assign a new range to an existing range name. To do so, specify the same name for oldName and newName and supply the new range as the range argument. SS_NAME_CHANGE@ should not be used to change range names for ranges that are linked to external spreadsheets. To change the names for ranges with external links, use SS_NAME_CHANGE_EXT@.

See also [SS_GET_NAMES@](#)
[SS_NAME_CHANGE_EXT@](#)
[SS_NAME_CREATE@](#)

SS_NAME_CHANGE_EXT@

Changes the alias name for an external spreadsheet link

Format SS_NAME_CHANGE_EXT@ (oldName, ssName, externalRangeName, newName, insertionPoint[, useLinkAttrFlag,] localizeLinkFlag])

Method [this.name_change_ext@ \(oldName, ssName, externalRangeName, newName, insertionPoint\[, useLinkAttrFlag, \] localizeLinkFlag \] \)](#)

Arguments

oldName	A string indicating the name by which the external range is currently identified in the current spreadsheet.
---------	--

ssName	A string giving the full path name of the external spreadsheet from which you want to reference data.
externalRangeName	A string indicating the name of the range to be referenced in the external document.
newName	A string indicating the name by which the external range will be identified in the current spreadsheet. Alias names can be from 3 to 30 characters, and the first three characters must be letters. Alias names cannot contain spaces. If newAlias is the same as an existing range name in the current spreadsheet, the existing name reference is overwritten.
insertionPoint	A string indicating the cell in the current spreadsheet where a copy of the external range should be placed.
useLinkAttrFlag	A Boolean value which if set to TRUE indicates that the linked information should be displayed using the attributes in the originating Spreadsheet.
localizeLinkFlag	A Boolean value which if set to TRUE indicates that external links are to be made local.

Description Change the range name in the current spreadsheet for a range that is a link to an external spreadsheet. For example, you could use `SS_NAME_CHANGE_EXT@` to change the range named "Profits_South" to the range named "Profits_Area2."

`SS_NAME_CHANGE_EXT@` should only be used to change range names for ranges that are linked to external spreadsheets. To change the names for ranges without external links, use [**SS_NAME_CHANGE@**](#).

See also [**SS_NAME_EXTERNAL@**](#)

SS_NAME_CREATE@

Assigns a name to a specified range

This macro is obsolete. Please use the macro [**SS_CREATE_NAMED_RANGE@**](#) for all new ELF programming.

Format `SS_NAME_CREATE@(name, range)`

Method `this.name_create@(name, range, refCell)`

Arguments name The name, a string, to assign to the range. If name already exists, an error is thrown. (To redefine an existing range, use the

SS_CHANGE_NAMED_RANGE@ macro.) Range names can be from 3 to 30 characters, and the first three characters must be letters. Range names cannot contain spaces.

range A string indicating the range to which name is assigned.
refCell If this argument exists, you are entering the address of a reference cell for a relative range.

Description Assigns a range name to the specified range in the current spreadsheet. Range names should not include the following spreadsheet reserved characters or words: + * ^ / ,) (> < = # . \$ ERROR, ERR, TRUE, FALSE, RAND, TODAY, and PI.

See also [SS_GET_NAMES@](#)
[SS_NAME_DELETE@](#)
[SS_NAME_CHANGE@](#)

SS_NAME_DELETE@

Deletes a range name

Format SS_NAME_DELETE@(name)

Method [this.name_delete@](#)(name)

Arguments **name** The name of the range being deleted. If name does not exist, an error is thrown.

Description Deletes a range name from the spreadsheets list of range names; it does not effect the contents of the range to which name refers in any way. Any cells that refer to a deleted range name will display ERROR.

See also [SS_GET_NAMES@](#)
[SS_NAME_CREATE@](#)

SS_NAME_EXTERNAL@

Creates a reference to an external spreadsheet

Format SS_NAME_EXTERNAL@ (externSSfileName, externalRange, alias, insertionPoint, useLinkAttrFlag)

Method `this.name_external@ (externSSfileName, externalRange, alias, insertionPoint, useLinkAttrFlag)`

Arguments

- externSSfileName**
A string giving the full path name of the external spreadsheet from which you want to reference data.
- externalRange**
A string indicating the name of the range to be referenced in the external document.
- alias**
A string indicating the name by which the external range will be identified in the current spreadsheet. Alias names can be from 3 to 30 characters, and the first three characters must be letters. Alias names cannot contain spaces.
If alias is the same as an existing range name in the current spreadsheet, the existing name reference is overwritten.
- insertionPoint**
A string indicating the cell in the current spreadsheet where a copy of the external range should be placed.
- useLinkAttrFlag**
A Boolean value which if set to TRUE indicates that the linked information should be displayed using the attributes in the originating Spreadsheet.

Description Creates a link in the current spreadsheet to a range in an external spreadsheet. The name specified by alias will appear in the current spreadsheet's external link name index. The cells in the copy of the external range that are displayed in the current spreadsheet are protected and therefore cannot be edited.

See also [**SS_NAME_CHANGE_EXT@**](#)

SS_NEW_SHEET@

Creates a new sheet

Format `SS_NEW_SHEET@()`

Method `this.new_sheet@`

Description Creates a new sheet in the current spreadsheet. The new sheet is inserted after existing sheets.

SS_NEW_WIN_ENV@

Displays a blank Spreadsheets window

Format SS_NEW_WIN_ENV@(id, pointer, mbFile)

Method [obj = this.new_win_env@\(id, pointer, mbFile\)](#)

Arguments

id	A user's unique identification number for this profile. id can be from 100 to 199.
pointer	A string indicating the image to use for the mouse pointer. For a list of possible symbol values, see SET LOOK FEEL VALUES@ . Set to inherit to make the pointer style default to the one specified in your ax_ss4 file.
mbFile	Name of the menu bar file to use with this window. The default is ax_ss4.

Description Displays a blank Spreadsheets window bearing the menu bar and mouse pointer of your choice. The custom menu bar and mouse pointer remain in memory throughout a session in a given window. By using SS_NEW_WIN_ENV@, you can display any number of different Spreadsheets menu bars at the same time.

SS_NEXT_SCREEN_KEY@

Display next screen of data

Format SS_NEXT_SCREEN_KEY@()

Method [this.next_screen_key@](#)

Description Displays the portion of the Spreadsheet document that is beneath the displayed portion.

See also [SS_PREV_SCREEN_KEY@](#)

SS_NUMBER_STYLE@

Sets the number display type for spreadsheet cell

Format SS_NUMBER_STYLE@(type, precision, range)

Method [this.number_style@\(type, precision, range\)](#)

Arguments	type	The number display type. The possible types are:
		0 unstyled
		1 Boolean
		2 general
		3 fixed
		4 scientific
		5 currency
		6 comma
		7 percentage
		8 date
		9 graph
		10 time
		11 Reserved for Future Use
		12 User-Defined
	precision	A number indicating the precision which numbers will be displayed in the cell. The default precision is 0, which means no decimal values. If, for example, you wanted to show numbers with two decimal values (such as 4.13), you would use a precision value of 2. For date types, the precision indicates the format in which the date is displayed. See the datetime.sp file for additional information on date and time formats.
	range	The range to which this style is being assigned.

Description Applies a number style to a spreadsheet cell. Number styles can be applied to empty cells. If a number style is applied to a cell containing a label or formula, the number style does not take effect.

A type value of 12 indicates a user-defined format is in effect. For more information on this feature, refer to Chapter 4 of the "Applixware Real-Time Guide."

See also [SS_GET_CELL@](#)
[SS_NUM_STYLE@](#)

SS_NUM_STYLE@

Sets the current selection's number display style

Format SS_NUM_STYLE@(type, precision)

Method [this.num_style@](#)(type, precision)

Arguments type The number display type. The possible types are:

0	unstyled
1	Boolean
2	general
3	fixed
4	scientific
5	currency
6	comma
7	percentage
8	date
9	graph

precision A number indicating the precision which numbers will be displayed in the cell. The default precision is 0, which means no decimal values. If, for example, you wanted to show numbers with two decimal values (such as 4.13), you would use a precision value of 2.

For date types, the precision indicates the format in which the date is displayed. See the [datetime.sp](#) file for additional information on date and time formats.

Description Applies a number style to the current selection. Number styles can be applied to empty selections. If a number style is applied to a cell containing a label or formula, the number style does not take effect.

See also [SS_GET_CELL@](#)
[SS_NUMBER_STYLE@](#)

SS_OBJECT_GET_NAMES@

Returns a list of Spreadsheet object names

Format nameArray = SS_OBJECT_GET_NAMES@()

Method nameArray = this.object_get_names@

Description Returns an array whose members are the names of objects contained within an Applixware Spreadsheets document.

SS_OBJECT_GET_TITLE@

Returns the title associated with an object

Format objectTitle = SS_OBJECT_GET_TITLE@(objectname)

Method objectTitle = this.object_get_title@(objectname)

Arguments objectname The name of a button object within the current Spreadsheets document.

Description Returns the title associated with the button object named objectname. This name was assigned to the object using [SS_OBJECT_SET_TITLE@](#).

SS_OBJECT_SET_TITLE@

Defines an object's title

Format SS_OBJECT_SET_TITLE@(objectName, title)

Method this.object_set_title@(objectName, title)

Arguments objectName The name of a button object within the current Spreadsheets document.
title The text that will become associated with a button object as its title.

Description Defines the text that will be associated with a button object as its title. Use [SS_OBJECT_GET_TITLE@](#) to obtain the current title string.

SS_OBJ_GET_LOCATION@

Returns the location of an object in the spreadsheet

Format format ss_object_loc@ loc = SS_OBJ_GET_LOCATION@(object)

Arguments object The name of an object within the spreadsheet.

Description Returns a formatted ELF array of type [ss_object_loc@](#). This array contains the cell location and offset information for the target object.

See also [SS_OBJ_SET_LOCATION@](#)

SS_OBJ_MOVE_BACK@

Moves an object behind all other objects

Format SS_OBJ_MOVE_BACK@(objName)

Method [this.obj_move_back@\(objName\)](#)

Arguments objName The name of the object in the spreadsheets document.

Description Moves an object behind all other objects.

SS_OBJ_MOVE_BACKWARD@

Moves an object in back of the previous object

Format SS_OBJ_MOVE_BACKWARD@(objName)

Method [this.obj_move_backward@\(objName\)](#)

Arguments objName The name of the object in the spreadsheets document.

Description Moves an object in back of the previous object.

SS_OBJ_MOVE_FORWARD@

Moves an object in front of next object

Format SS_OBJ_MOVE_FORWARD@(objName)

Method [this.obj_move_forward@\(objName\)](#)

Arguments objName The name of the object in the spreadsheets document.

Description Moves an object in front of next object.

SS_OBJ_MOVE_FRONT@

Places an object in front of all other objects

Format SS_OBJ_MOVE_FRONT@(objName)

Method [this.obj_move_front@\(objName\)](#)

Arguments objName The name of the object in the spreadsheets document.

Description Places an object in front of all other objects.

SS_OBJ_SET_LOCATION@

Sets the position of an object
in the Spreadsheet

Format SS_OBJ_SET_LOCATION@(format ss_object_loc@ location)

Arguments location A formatted ELF array of type [ss_object_loc@](#).

Description Sets the location of an object in the current sheet.

SS_OPEN_CELL_AS_STRING@

Returns the cell address of a cell

Format addressString = SS_OPEN_CELL_AS_STRING@()

Method [addressString = this.open_cell_as_string@](#)

Description Returns the string representation of the current cell.

SS_OPEN_OBJECT@

Opens a graphic or chart object

Format SS_OPEN_OBJECT@(name)

Method [this.open_object@\(name\)](#)

Arguments name The internal name of the chart or graphics object.

Description Opens an object either linked to or stored within the Spreadsheet document.

See also [SS_CLOSE_OBJECTS@](#)

SS_PASTE@

Pastes the contents of the clipboard at the current cursor location

Format SS_PASTE@(firstCell, [unused], valuesOnly)

Method [this.paste@](#)(firstCell, [unused], valuesOnly)

Arguments

firstCell	The address (as a string) of the first cell into which the clipboard is being pasted.
unused	Set to NULL.
valuesOnly	Indicates whether formulas should be pasted along with the cells' values. TRUE indicates that only the values are being pasted; FALSE indicates that both values and formulas (if any) are being pasted.

Description Pastes the contents of the clipboard at the current cursor position in the current spreadsheet. SS_PASTE@ is called by the Edit ® Paste menu option.

See also [SS_COPY@](#)

[SS_CUT@](#)

[SS_PASTE_SPECIAL@](#)

SS_PASTE_IMPORT@

Imports an ASCII file into a Spreadsheets window, starting at the current cell

Format SS_PASTE_IMPORT@(file, asciiType, window)

Method [this.paste_import@](#)(file, asciiType, window)

Arguments

file	The full path name (as a string) of the ASCII file to import.
asciiType	Indicates how line breaks and other delimiters in the ASCII file should be interpreted when filling cells in the spreadsheet:

- l Imports and pastes a WK1 file.
- s Imports and pastes a SYLK file.
- 9 Grid style: Every line break in the ASCII file translates to a row break in the spreadsheet; every delimiter (as defined in * ® Spreadsheets Preferences) in the ASCII file translates to a column break in the spreadsheet. See [SS_IMPORT_GRID@](#).
- 10 Column style: Fills only one column. Place each ASCII line into one cell; advance to the next column with every line break in the ASCII file. For more information, see [SS_IMPORT_COL@](#).
- 11 Row style: Fills only one row. Place each ASCII line into one cell; advance to the next row with every line break in the ASCII file. See [SS_IMPORT_ROW@](#).

window The path name of the Spreadsheets document into which you want to paste the ASCII file. window must be an open window (or at least an icon), but it does not have to be the current window.

Description Imports the specified ASCII file into the specified spreadsheet, starting from the cell at cursor position.

See also [SS_PASTE_IMPORT_CURDOC@](#)

SS_PASTE_IMPORT_CURDOC@

Imports an ASCII file into a Spreadsheets window

Format SS_PASTE_IMPORT_CURDOC@(file, asciiType)

Method [this.paste_import_curdoc@\(file, asciiType\)](#)

Arguments

file	The full path name (as a string) of the ASCII file to import.
asciiType	Indicates how line breaks and other delimiters in the ASCII file should be interpreted when filling cells in the spreadsheet: <ul style="list-style-type: none"> l Imports and pastes a WK1 file. s Imports and pastes a SYLK file. 9 Grid style: Every line break in the ASCII file translates to a row break in the spreadsheet; every delimiter (as defined in * ® Spreadsheets Preferences) in the ASCII file translates to a column break in the spreadsheet. See SS_IMPORT_GRID@.

- 10 Column style: Fills only one column. Place each ASCII line into one cell; advance to the next column with every line break in the ASCII file. For more information, see [SS_IMPORT_COL@](#).
- 11 Row style: Fills only one row. Place each ASCII line into one cell; advance to the next row with every line break in the ASCII file. See [SS_IMPORT_ROW@](#).

Description Imports an ASCII file into the current spreadsheet, starting from the current cell without deleting any data that exists in the spreadsheet (the information is inserted into the spreadsheet using [SS_PASTE@](#))

See also [SS_PASTE_DLG@](#)

SS_PASTE_SPECIAL@

Pastes into a cell

Format SS_PASTE_SPECIAL@(dest, pasteType)

Method [this.paste_special@](#)(dest, pasteType)

Arguments

dest	The place into which the information will be pasted.
pasteType	A two_element array argument indicating the paste type. pasteType[0] is one of the following values:
0	all
1	formulas
2	styles
3	values

Pastetype[1] is a number from 0 to 3, which determines whether the data is transposed. Possible values are as follows:

0	no transposition.
1	Rows/Columns. Rearranges the pasted material so that columns and rows are interchanged. If the range on the clipboard was organized by rows, the pasted range is organized by columns. If the range on the clipboard was organized by columns, the pasted range is organized by rows. The number of rows becomes the number of columns and the number of columns becomes the number of rows.

If a 3D range is selected, the rows and columns on each sheet are interchanged.

- 2 Sheets / Rows. Rearranges the pasted material so that sheets and rows are interchanged. The first row of the range in the first sheet contains the first row of data. The same row of the range in the second sheet contains the second row of data. The same row of the range in the third sheet contains the third row of data and so on. If a 3D range was selected, the first row of the range in the first sheet contains the first row of data. The second row of the range in the second sheet contains the second row of data. The third row of the range in the third sheet contains the third row of data and so on.
- 3 Sheets / Columns. Rearranges the pasted material so that sheets and columns are interchanged. The first column of the range in the first sheet contains the first column of data. The same column of the range in the second sheet contains the second column of data. The same column of the range in the third sheet contains the third column of data and so on. If a 3D range was selected, the first column of the range in the first sheet contains the first column of data, the second column of the range in the second sheet contains the second column of data, the third column of the range in the third sheet contains the third column of data and so on.

Description Pastes any combination of cell values, styles or formulas from the clipboard into your spreadsheet, or transposes columns and rows when pasting.

See also [SS_PASTE@](#)

SS_PREV_SCREEN_KEY@

Moves one screen back vertically

Format SS_PREV_SCREEN_KEY@()

Method [this.prev_screen_key@](#)

Description Moves one screen back vertically like a Page Up.

See also [SS_NEXT_SCREEN_KEY@](#)

SS_PREVIEW@

Shows how the spreadsheet will look when printed

Format SS_PREVIEW@()

Method [this.preview@](#)

Description Creates a new display window and draws an image of how the current spreadsheet will look when it is printed.

For general information, see [Print Preview](#).

SS_PREVIEW_CLOSE@

Closes the preview window

Format SS_PREVIEW_CLOSE@()

Method [this.preview_close@](#)

Description Closes the *Print Preview* window.

For general information, see **File** ® [Exit](#).

SS_PREVIEW_MARGINS@

Hides or displays the margins

Format SS_PREVIEW_MARGINS@()

Method [this.preview_margins@](#)

Description Allows you to adjust the spreadsheets margins from within the Print Preview window.

For more information, see **View** ® [Margins](#).

SS_PREVIEW_NEXT_PAGE@

Scrolls to the next page

Format SS_PREVIEW_NEXT_PAGE@()

Method [this.preview_next_page@](#)

Description Moves to the next page in the view displayed in the Print Preview dialog box.
For more information, see **View ® Page ® [Next](#)**.

SS_PREVIEW_NEXT_VIEW@

Shows the new view

Format SS_PREVIEW_NEXT_VIEW@()

Method [this.preview_next_view@](#)

Description Displays the next view selected in the Print dialog box.
For more information, see **View ® Named View ® [Next](#)**.

SS_PREVIEW_PREV_PAGE@

Shows the previous page

Format SS_PREVIEW_PREV_PAGE@()

Method [this.preview_prev_page@](#)

Description Moves to the preceding page in the view displayed in the Print Preview dialog box.
For more information, see **View ® Page ® [Previous](#)**.

SS_PREVIEW_PREV_VIEW@

Show the previous view

Format SS_PREVIEW_PREV_VIEW@()

Method [this.preview_prev_view@](#)

Description Displays the preview view selected in the Print dialog box. This command can only be invoked when you access Print Preview from within the Print dialog box.

For more information, see **View** ® **Named View** ® [Previous](#).

SS_PREVIEW_REFORMAT@

Reformats the information in the preview window

Format SS_PREVIEW_REFORMAT@()

Method [this.preview_reformat@](#)

Description Updates the Print Preview window so that it reflects the current state of the spreadsheet. The Print Preview screen does not dynamically update, and will not display a sheet or view which was not visible in the spreadsheet window or selected in the Print dialog box.

SS_PREVIEW_REPAINT@

Redraws the preview window

Format SS_PREVIEW_REPAINT@()

Method [this.preview_repaint@](#)

SS_PREVIEW_TRUE_ZOOM@

Zooms (or unzooms) the preview window

Format SS_PREVIEW_TRUE_ZOOM@()

Method [this.preview_true_zoom@](#)

Description If the preview window is not zoomed, this macro zooms the scale to 100%. If it is already being displayed at 100%, the image is scaled so that it fits within the preview window.

For more information, see **Utilities** ® [Zoom](#).

SS_PRINT@

Prints the current spreadsheet

Format SS_PRINT@ (printer, colorFlag, copies, bannerFlag, allPagesFlag, startPage, endPage, NULL, viewname, NULL, tempFile[,printFormulasFlag][, PgExt1][, PgExt2] [,backgroundFlag][, rangeStr][, hiddenAlsoFlag][, class][, baggage])

Method [this.print@](#) (printer, colorFlag, copies, bannerFlag, allPagesFlag, startPage, endPage, NULL, viewname, NULL, tempFile[,printFormulasFlag][, PgExt1][, PgExt2] [,backgroundFlag][, rangeStr][, hiddenAlsoFlag][, class][, baggage])

Arguments

printer	A string giving the name of the printer on which to print the document. If you want to print to a file rather than to a printer, set to NULL; the print file is placed in the directory specified by tempFile, or in the Applixware temporary directory if no directory is specified with tempFile.
colorFlag	Indicates whether the document is being printed on a color printer. If you are printing on a color printer, set color to TRUE; otherwise, set color to FALSE. Default is FALSE.
copies	The number of copies to print. The default is 1.
bannerFlag	Indicates whether to include a banner page with the printed document. Set to TRUE if you want a banner page, FALSE if you don't. The default is FALSE.
allPagesFlag	Whether to print all pages of the document. Set to TRUE to print all pages, FALSE if you want to print a range of pages.
startPage	A number indicating the first page to print if you are printing a range of pages. If allPagesFlag is TRUE, startPage is ignored.
endPage	A number indicating the last page to print if you are printing a range of pages. If allPagesFlag is TRUE, endPage is ignored.
viewname	The view name of the spreadsheet, if any.
tempFile	The name of the print file. This file is created to print the document and is deleted when printing is completed (unless printer is set to NULL). If you do not supply a path name for tempFile, the print file is created in the Applixware temporary directory.
printFormulasFlag	Set to TRUE to print formulas; set to NULL to print the document.

PgExt1	Indicates the beginning page of any <i>lateral</i> pages to be printed. For example, if you want to print from Page 1a to 2d, type 1. Set to NULL if no lateral pages are included.
PgExt2	Indicates the ending page of any <i>lateral</i> pages to be printed. For example, if you want to print from Page 1a to 2d, type 4. Set to NULL if no lateral pages are included.
backgroundFlag	Indicates whether to print the document in the ``background." Set to FALSE to not print in the background. If set to TRUE, a new axmain (an additional Applixware server process) will be created, without checking out a new license. That new process will be killed upon completing the print job.
rangeStr	A range to be printed. This parameter is ignored during background printing.
hiddenAlsoFlag	A Boolean value which if set to TRUE indicates that invisible information should also be printed. This parameter is ignored during background printing.
class	A constant indicating if printing will be performed on a PostScript or PCL device, as follows: PostScript PCL5
baggage	an array of format print baggage@ .

Description Prints the current Spreadsheets document using the attributes you specify. It is best to assign a FALSE value to the background argument unless you are printing a very long document (approximately 20 pages or more). Otherwise, it may take longer to invoke the new Applixware process than to print in the foreground.

SS_PRINT_DOC_FORMULAS@

Prints only the formulas in a document

Format SS_PRINT_DOC_FORMULAS@(printer, colorFlag, copies, bannerFlag, allPagesFlag, startPage, endPage, pgExt1, pgExt2, NULL, viewName, tempFile[, rangeStr[, hiddenFlag[, class]]])

Method [this.print_doc_formulas@](#)(printer, colorFlag, copies, bannerFlag, allPagesFlag, startPage, endPage, pgExt1, pgExt2, NULL, viewName, tempFile[, rangeStr[, hiddenFlag[, class]]])

Arguments	printer	A string giving the name of the printer on which to print the document. If you want to print to a file rather than to a printer, set to NULL; the print file is placed in the directory specified by tempFile, or in the Applixware temporary directory if no directory is specified with tempFile.
	colorFlag	Indicates whether the document is being printed on a color printer. If you are printing on a color printer, set color to TRUE; otherwise, set color to FALSE. The default is FALSE.
	copies	The number of copies to print. The default is 1.
	bannerFlag	Indicates whether to include a banner page with the printed document. Set to TRUE if you want a banner page, FALSE if you don't. Default is FALSE.
	allPagesFlag	Whether to print all pages of the document. Set to TRUE to print all pages, FALSE if you want to print a range of pages.
	startPage	A number indicating the first page to print if you are printing a range of pages. If allPagesFlag is TRUE, startPage is ignored.
	endPage	A number indicating the last page to print if you are printing a range of pages. If allPagesFlag is TRUE, endPage is ignored.
	PgExt1	Indicates the beginning page of any <i>lateral</i> pages to be printed. For example, if you want to print from Page 1a to 2b, type 1a. Set to NULL if no lateral pages are included.
	PgExt2	Indicates the ending page of any <i>lateral</i> pages to be printed. For example, if you want to print from Page 1a to 2b, type 2b. Set to NULL if no lateral pages are included.
	viewName	The view name of the spreadsheet, if any.
	tempFile	The name of the print file. This file is created to print the document and is deleted when printing is completed (unless printer is set to NULL). If you do not supply a path name for tempFile, the print file is created in the Applixware temporary directory.
	rangeStr	A range, or set of ranges, to be printed.
	hiddenFlag	A Boolean value which if set to TRUE indicates that invisible information should also be printed.
	class	A constant indicating if printing will be performed on a PostScript or PCL device, as follows: PostScript PCL5

Description Prints the current Spreadsheets document using the attributes you specify. When the Spreadsheets is printed, it will print a list of the formulas referenced in the cells.

SS_PROJ_TABLE1@

Builds a one-variable projection table

Format SS_PROJ_TABLE1@(input, output, range)

Method [this.proj_table1@\(input, output, range\)](#)

Arguments

input	A string indicating the cell address for the input cell for the projection table. For example, "a1" indicates that cell A1 will contain the input values for your projection table.
output	A string indicating the cell address for the output cell for the projection table. The output cell contains the formula used to make calculations based on the value of input.
range	A string indicating a two-column range into which the input and output values are written.

Description Creates a projection table is created by calculating the formula in output using the indicated input values. The result of the calculation is placed into range.

The input cell must have a value when SS_PROJ_TABLE1@ is called. The value of input cannot be a formula.

See also [SS_PROJ_TABLE2@](#)
[Using Projection Tables](#)

SS_PROJ_TABLE2@

Builds a two-variable projection table

Format SS_PROJ_TABLE2@(input1, input2, output, range)

Method [this.proj_table2@\(input1, input2, output, range\)](#)

Arguments

input1	A string indicating the cell address for the projection table's first input cell.
input2	A string indicating the cell address for the projection table's second input cell.

output	A string indicating the projection table's output cell address. This cell contains the formula used to make calculations based on the value of input1 and input2.
range	A string indicating a range address into which the input and output values are written.

Description The first column and first row in range should contain input values. The cell at the intersection of the input row and column must be empty. The projection table is created by calculating the formula in output using the two variable input values specified by input1 and input2.

The cells specified by input1 and input2 must have values when SS_PROJ_TABLE2@ is called. The values of input1 and input2 cannot be formulas.

See also [SS_PROJ_TABLE1@](#)
[Using Projection Tables](#)

SS_PROTECT@

Toggles a selection's protection state

Format SS_PROTECT@()

Method [this.protect@](#)

Description Changes the state of the selected range from unprotected to protected (or from protected to unprotected.)

When cells are protected, they cannot be edited or deleted.

SS_PUT_ARRAY_FORMULA@

Inserts an array of values

Format SS_PUT_ARRAY_FORMULA@(range, value[, forceFlag], [redisplayflag])

Method [this.put_array_formula@\(range, value\[, forceFlag \]\)](#)

Arguments

range	The range into which values are inserted.
formula	A string containing a formula that returns an array.
forceFlag	Indicates whether to enter the value even if the cell is protected. Specify TRUE if to make the entry regardless of the cell's protection; specify

FALSE, which is the default, if the cell should not be edited if it is protected. FALSE triggers an error message if the cell is protected.

redisplayFlag Indicates whether to refresh the screen display when the array formula is updated. The default is FALSE.

Description Inserts an [array formula](#) into a range of Spreadsheet cells. This macro allows you to insert all values returned by the formula into the cells of a range. If the range is larger than the array returned by the formula, the remaining cells are padded with NA values.

SS_PUT_CELL@

Enters a value into a cell

Format SS_PUT_CELL@(cell, value[,forceFlag[, redisplayFlag]])

Method [this.put_cell@](#)(cell, value[,forceFlag[, redisplayFlag]])

Arguments

cell	A cell address (as a string).
value	The value (as a string) to be entered into the cell.
forceFlag	Indicates whether to enter the value even if the cell is protected. Specify TRUE if you want to make the entry regardless of the cell's protection. Specify FALSE if the cell should not be edited if it is protected. FALSE triggers an error message if the cell is protected. The default is FALSE.
redisplayFlag	Indicates whether to refresh the screen display when value is entered into cell. The default is FALSE.

Description Places a string value into a cell. This macro is most efficient when forceFlag is omitted or set to FALSE before entering a series of values. You would then set redisplayFlag to TRUE after all values in this series are entered.

See also [SS_GET_CELL_VALUE@](#)

SS_PUT_RANGE@

Places an array of data into a sheet

Format SS_PUT_RANGE@(range, data[, ignoreNullsFlag[, forceFlag[, updateDisplayFlag]]])

Method [this.put_range@](#)(range, data[, ignoreNullsFlag[, forceFlag[, updateDisplayFlag]]])

Arguments	range	The range into which the data will be inserted.
	data	The array of information being inserted into the spreadsheet.
	ignoreNullsFlag	A Boolean value where TRUE indicates that NULL array values are ignored. Setting flag to TRUE can cause a range to not be completely filled when data is inserted.
	forceFlag	A Boolean value where TRUE indicates that data can be inserted into protected cells.
	updateDisplayFlag	A Boolean value where TRUE indicates that the position and display are updated after information is inserted.

Description Inserts an array of data into a Spreadsheets document.

SS_RANGE@

Returns a range string based on the numbers supplied

Format range = SS_RANGE@(leftCol, topRow, rightCol, bottomRow[, startSheet[, endSheet]])

Method range = this.range@(leftCol, topRow, rightCol, bottomRow[, startSheet[, endSheet]])

Arguments	leftCol	A number representing the column for the top left cell in the range. Columns are numbered from 0, with column A being 0, column B being 1, and so on.
	topRow	A number representing the row for the top left cell in the range. Rows are numbered from 0, with row 1 being 0, row 2 being 1, and so on.
	rightCol	A number representing the column for the bottom right cell in the range.
	bottomRow	A number representing the row for the bottom right cell in the range.
	startSheet	A number representing the sheet for the lowest numbered sheet in the range. Sheets are numbered from 0, with sheet A being 0, sheet B being 1, and so on. If no startSheet is specified, the current sheet is used.
	endSheet	A number representing the last sheet in the range. If no endSheet is specified, the current sheet is used.

Description Converts the numbers provided to a range string. For example, SS_RANGE@(0, 1, 5, 8, 0, 5) returns the range string A2..F9.

See also [SS_DELETE_RANGE@](#)
[SS_NAME_CREATE@](#)

SS_RECALC@

Recalculates the current spreadsheet

Format SS_RECALC@()

Method [this.recalc@](#)

Description Recalculates a spreadsheet that is in manual calculation mode. SS_RECALC@ is called by the Edit ® Recalculate menu option.

See also [SS_CALC@](#)

SS_RECORD_MACRO@

Records a Spreadsheets macro

Format SS_RECORD_MACRO@()

Method [this.record_macro@](#)

SS_REDISPLAY@

Updates the screen

Format SS_REDISPLAY@()

Method [this.redisplay@](#)

Description Updates the screen display. Updating differs from repainting in that updating insures that all cell values are up-to-date. Repainting simply clears the screen and redraws it. (See [SS_REPAINT_WINDOW@](#).)

SS_REFERENCES_CELL@

Lists all references or dependents of a cell

Format SS_REFERENCES_CELL@(cell[, typeFlag[, allFlag]])

Method [this.references_cell@\(cell\[, typeFlag\[, allFlag \] \]\)](#)

Arguments	cell	The cell for which references or dependents will be listed.
	typeFlag	A Boolean value which if set to TRUE indicates that references are listed. If it is set to FALSE, dependents are listed. The default is FALSE.
	allFlag	A Boolean value with if set to TRUE indicates that all references or dependents are returned. The default is FALSE.

Description Returns either all references or dependents of a cell or the immediate reference or dependent of the cell.

- The cells to which the current cell refers to are called *reference* cells.
- The cells which refer to (that is, depend on) the current cell are called *dependent* cells.

SS_REGISTER_FUNCTION@

Adds a user function to the Spreadsheets functions list

Format SS_REGISTER_FUNCTION@(category, registerName, argString, [transString])

Method [this.register_function@](#)(category, registerName, argString, transString)

Arguments	category	The function's category. This string appears in the Categories box when you select Tools ® Functions from within Spreadsheets.
	registerName	This is the name that you use to register the function with Applixware. To unregister the function using the macro SS_UNREGISTER_FUNCTION@ , you must also use this name.
	argString	This string is displayed in the Functions box when you select Tools ® Functions from within Spreadsheets. By convention, the string contains the name of the function and its arguments. If the function has more than one argument, the arguments should be separated with a comma followed by a space.
	transString	The function name that is saved to the .as file when you are running a language other than english. If you are running English, this is an optional parameter. If you are running another language, this parameter is required.

Description Adds a user-defined function to the Spreadsheets built-in function lists. The name of the macro specified in the argstring argument must correspond to a macro file in your search path. For example, if the argString is foo(a,b), the file foo.am must be in your search path.

The translation string argument applies only if you are running a language other than english. Suppose you register a function as follows:

SS_REGISTER@("Financial", "COUNT", "COUNT(A,B)", "Compter(A,B)")

In an English version of Applixware, this function would be written to disk as "+COUNT(A;B)". In a French version of Applixware, this function would be written to disk as "+COMPTER(A;B)".

SS_RENAME@

Renames the current Spreadsheet to a new name

Format SS_RENAME@(infoArray)

Method [this.rename@](#)(infoArray)

Arguments infoArray A four element array defined as follows:

name	The current file's new name.
mode	The save mode
	1 (The default:) Saved in machine independent (portable) format
	2 Saved in Lotus WK1 format
	3 Saved in SYLK format
	4 Saved in DIF format
	5 Saved in CSV format
	6 Saved in ASCII Grid format
	7 Saved in ASCII Column format
	8 Saved in ASCII Row format
	9 Saved in Lotus WK3 format
	10 Saved in XLS version 3.0 format
	11 Saved in Aster*x 2.1 format
	12 Saved in XLS version 4.0 format
groupAccess	A number indicating the read and write permissions for the file for a member of the same group. This can be one of the following values:
	0 No read or write permissions for the file
	1 Read permissions for the file
	2 Read and write permissions for the file
allAccess	A number indicating the read and write permissions for any user. This can be one of the following values.
	0 No read or write permissions for the file
	1 Read permissions for the file
	2 Read and write permissions for the file

Description Renames a Spreadsheet document using the attributes passed to its info argument.

See also [SS_SAVE_AS@](#)

SS_RENAME_CHART@

Changes a chart's name

Format SS_RENAME_CHART@(oldName, newName)

Method [this.rename_chart@](#)(oldName, newName)

Arguments oldName The name of the chart whose name will be changed.
newName The name to which a chart will be renamed.

Description Changes a chart's name from oldName to newName.

See also [SS_GET_CHARTS_IN_DOC@](#)

SS_RENAME_OBJECT@

Changes an object's name

Format SS_RENAME_OBJECT@(oldName, newName)

Method [this.rename_object@](#)(oldName, newName)

Arguments oldName The name of the object whose name will be changed
newName The name to which an object will be renamed.

Description Changes an object's name from oldName to newName.

SS_REPAINT_WINDOW@

Repaints the Spreadsheets window

Format SS_REPAINT_WINDOW@()

Method [this.repaint_window@](#)

See also [SS_REDISPLAY@](#)

SS_RETURN_KEY@

Moves the cell pointer to the next cell

Format SS_RETURN_KEY@()

Method [this.return_key@](#)

See also [SS_BACK_RETURN_KEY@](#)
[SS_DOWN_ARROW_KEY@](#)
[SS_LEFT_ARROW_KEY@](#)
[SS_RIGHT_ARROW_KEY@](#)
[SS_UP_ARROW_KEY@](#)

SS_REVEAL@

Reveals all hidden cells in the current spreadsheet

Format SS_REVEAL@()

Method [this.reveal@](#)

Description Displays all spreadsheet cells that have previously been hidden. SS_REVEAL@ is called by the View ® View All menu option.

See also [SS_HIDE@](#)
[SS_REVEAL_SOME@](#)

SS_REVEAL_SOME@

Reveals the specified rows and/or columns

Format SS_REVEAL_SOME@(list)

Method [this.reveal_some@\(list\)](#)

Arguments list A string indicating the rows and columns to reveal and select. Each row number or column letter in list is separated by a comma. For example, to

reveal and select columns E and F and rows 6 and 8, specify list as "E,F,6,8". A range of rows or columns can be specified in the form: beginning range-end range.

Description Reveals rows and columns that have previously been hidden.

See also [SS_HIDE@](#)
[SS_REVEAL@](#)

SS_REVERT@

Restores a Spreadsheets document

Format SS_REVERT@()

Method [this.revert@](#)

Description Restores a Spreadsheets document to its state when last saved. This removes all changes made to the document since the last save. The document remains open. Unlike the File ® Revert menu option, SS_REVERT@ does not prompt to verify that the user really wants to restore the document.

SS_RIGHT_ARROW_KEY@

Moves the cell pointer right one cell

Format SS_RIGHT_ARROW_KEY@()

Method [this.right_arrow_key@](#)

See also [SS_BACK_RETURN_KEY@](#)
[SS_DOWN_ARROW_KEY@](#)
[SS_LEFT_ARROW_KEY@](#)
[SS_RETURN_KEY@](#)
[SS_UP_ARROW_KEY@](#)

SS_RIGHT_SCREEN_KEY@

Displays information to the right

Format SS_RIGHT_SCREEN_KEY@()

Method [this.right_screen_key@](#)

Description Displays information to the right of the current screen.

See also [SS_LEFT_SCREEN_KEY@](#)
[SS_TAB_KEY@](#)

SS_RIGHT_SECTION@

Moves the cursor to the right to the next cell containing data

Format SS_RIGHT_SECTION@()

Method [this.right_section@](#)

Description Moves the cursor from its current position to the next cell to the right that contains data. If no cells containing data are found to the right of the current cursor position, the cursor is moved to the last cell in the row. SS_RIGHT_SECTION@ is called by the Keys ® Next data right menu option.

See also [SS_LEFT_SECTION@](#)

SS_RNGSTR_TO_ABS_RNGSTR@

Converts relative range to absolute range

Format absString = SS_RNGSTR_TO_ABS_RNGSTR@(rangeString)

Method [absString = this.rngstr_to_abs_rngstr@\(rangeString\)](#)

Arguments rangeString The range string being converted.

Description Converts a range that is expressed as a relative row/column reference into a reference that is expressed as an absolute row/column address.

SS_RT_ALLOW_LIVE_TOGGLE@

Toggles the feed

Format SS_RT_ALLOW_LIVE_TOGGLE@()

Method [this.rt_allow_live_toggle@](#)

Description Toggles the state of the Real Time feed so that if it is sending data, data will no longer be sent. On the other hand, if data is not being sent, this macro tells the Real Time system that it is alright to start sending data.

See also [SS_RT_LIVE_ENABLE@](#) and [SS_RT_LIVE_DISABLE@](#).

SS_RT_LIVE_DISABLE@

Stops information from being sent

Format SS_RT_LIVE_DISABLE@()

Method [this.rt_live_disable@](#)

Description Tells the Real Time feed that it should not send any more data until it receives an [SS_RT_LIVE_ENABLE@](#) command.

SS_RT_LIVE_ENABLE@

Starts information flowing if currently disabled

Format SS_RT_LIVE_ENABLE@()

Method [this.rt_live_enable@](#)

Description Tells the Real Time feed that it should send data. Data will continue to be send until the Real Time feed receives an [SS_RT_LIVE_DISABLE@](#) command.

SS_RT_LIVE_FEED_STATUS@

Returns TRUE if Real Time Live Feed is enabled

Format boolean = SS_RT_LIVE_FEED_STATUS@()

Method [this.ss_rt_live_feed_status@](#)

Description Returns TRUE if the Utilities ® real time ® Enable live feed feature is enabled. Returns FALSE if the feature is disabled.

SS_SAVE@

Saves the current Spreadsheets document

Format SS_SAVE@()

Method [this.save@](#)

Description Writes the current Spreadsheets document to a file. If the spreadsheet has never been saved, it is saved to a file having the name displayed in the Spreadsheets window title area.

SS_SAVE_AS@

Saves a Spreadsheets document using specified attributes

Format SS_SAVE_AS@(name[, mode[, grpAcces[, allAccess]]])

Method [this.save_as@\(name\[, mode\[, grpAcces\[, allAccess\] \] \]\)](#)

Arguments

name	The full path name of the document to be saved, including the .as extension. If name already exists, the existing file is replaced with this new one without any warning.
mode	The mode in which to save the document. The modes are: 1 (The default:) Saved in machine independent (portable) format 2 Saved in Lotus WK1 format

3	Saved in SYLK format
4	Saved in DIF format
5	Saved in CSV format
6	Saved in ASCII Grid format
7	Saved in ASCII Column format
8	Saved in ASCII Row format
9	Saved in Lotus WK3 format
10	Saved in XLS version 3.0 format
11	Saved in Aster*x 2.1 format <i>This mode is obsolete and will not be supported in the future</i>
12	Saved in XLS version 4.0 format
13	Save in Applixware 3.11 format
14	Saved in XLS version 5.0 format
grpAccess	A number indicating the read and write permissions for the file for a member of the same group. grpAccess can be one of the following: 0 (The default:) No read or write permission for the file. 1 Read permission for the file. 2 Read and write permission for the file.
allAccess	A number indicating the read and write permissions for the file for any user. allAccess can be 0, 1, or 2, as described above. The default is 0.

See also [SS_RENAME@](#)

SS_SAVE_PROFILE@

Saves the current spreadsheet preferences

Format SS_SAVE_PROFILE@()

Method [this.save_profile@](#)

Description Saves the ax_prof4 file. This will save all current spreadsheet preference settings as well as many general Applixware settings.

SS_SAVE_VIEW@

Assigns a view name to the current display

Format SS_SAVE_VIEW@(name[, widthChangesFlag[, pageBreaksFlag[, heightsFlag]]])

Method [this.save_view@\(name\[, widthChangesFlag\[, pageBreaksFlag\[, heightsFlag \] \] \]\)](#)

Arguments	name	The name, a string, to save the view as. If name is the same as an existing view name, the existing view is overwritten by this new view.
	widthChangesFlag	Indicates whether column width changes are propagated in the newly created view. If widthChangesFlag is TRUE, any width changes to columns in other views will be applied to this new view if the column is contained in the view. If widthChangesFlag is FALSE, width changes in other views do not affect the newly created view.
	pageBreaksFlag	Indicates whether page breaks are propagated in the newly created view. If pageBreaksFlag is TRUE, any page break changes made to other views are applied to the new view. If pageBreaksFlag is FALSE, page break changes made to other views do not affect the newly created view.
	heightsFlag	A Boolean value which if set to TRUE indicates that row height changes are propagated in the newly created view. If this value is FALSE, row height changes in other view do not affect this newly created view.

Description Saves a view with the name and propagation attributes specified.

SS_SEARCH@

Searches for the specified label

Format SS_SEARCH@(label[, backFlag])

Method [this.search@](#)(label[, backFlag])

Arguments	label	A string indicating the label to search for.
	backFlag	If TRUE, the spreadsheet is searched from the current cursor location to the beginning of the spreadsheet for label. If FALSE (which is the default), the spreadsheet is searched from the current cursor location to the end of the spreadsheet for label.

Description Moves the cursor to the next or previous occurrence of the label specified by label. SS_SEARCH@ is case-insensitive and will find label if it is a whole word or a substring.

NOTE: If `SS_SEARCH@` does not find the specified label, it throws an ``info" error message that cannot be trapped by error number or error string. It is best to include an error handler that re-directs the flow of the macro appropriately.

See also [SS_SEARCH_NEXT@](#)
[SS_SEARCH_PREV@](#)

SS_SEARCH_NEXT@

Searches for the next occurrence of previously specified label

Format `SS_SEARCH_NEXT@()`

Method `this.search_next@`

Description Searches forward from the current cursor position in the spreadsheet. If another occurrence of the search string is not found, an error is thrown.

`SS_SEARCH_NEXT@` is called by the Find ® Text ® Next menu option.

See also [SS_SEARCH@](#)
[SS_SEARCH_PREV@](#)

SS_SEARCH_NEXT_SPECIAL@

Searches for the next occurrence of the previously specified status message

Format `SS_SEARCH_NEXT_SPECIAL@()`

Method `this.search_next_special@`

Description Searches forward from the current cursor position in the spreadsheet. If another occurrence of the status message is not found, an error is thrown.

`SS_SEARCH_NEXT_SPECIAL@` is called by the Find ® Status ® Next menu option.

See also [SS_SEARCH_PREV_SPECIAL@](#)
[SS_SEARCH_SPECIAL@](#)

SS_SEARCH_PREV@

Searches for the previous occurrence of a label

Format SS_SEARCH_PREV@()

Method [this.search_prev@](#)

Description Searches backwards from the current cursor position in the spreadsheet. If another occurrence of the search string is not found, an error is thrown. SS_SEARCH_PREV@ is called by the Find ® Text ® Previous menu option.

See also [SS_SEARCH@](#)
[SS_SEARCH_NEXT@](#)

SS_SEARCH_PREV_SPECIAL@

Searches for the previous occurrence of a status message

Format SS_SEARCH_PREV_SPECIAL@()

Method [this.search_prev_special@](#)

Description Searches backwards from the current cursor position in the spreadsheet. If another occurrence of the status message is not found, an error is thrown.

SS_SEARCH_PREV_SPECIAL@ is called by the Find ® Status ® Previous menu option.

See also [SS_SEARCH_NEXT_SPECIAL@](#)
[SS_SEARCH_SPECIAL@](#)

SS_SEARCH_SPECIAL@

Searches for a status message

Format SS_SEARCH_SPECIAL@(status[, backFlag])

Method [this.search_special@\(status\[, backFlag \]\)](#)

Arguments status The status message to search for. status is a string and can be one of the following:

0	ERROR
1	NA
2	CIRCULAR
3	OBSOLETE
4	PENDING
5	ALL (any of the above)

backFlag If TRUE, the spreadsheet is searched from the current cursor location to the beginning of the spreadsheet for status. If FALSE, the spreadsheet is searched from the current cursor location to the end of the spreadsheet for status.

Description Moves the cursor to the next or previous occurrence of the status message specified by status.

See also [SS_SEARCH_NEXT_SPECIAL@](#)
[SS_SEARCH_PREV_SPECIAL@](#)

SS_SELECT@

Returns the addresses or other information for selected cells

Format info = SS_SELECT@([type[, pos]])

Method info = this.select@([type[, pos]])

Arguments type A string that specifies the criteria for the cell addresses or other information that will be returned. If you do not include type, the cell address of all selected cells is returned. type can be any of the following:

1	All selected cells are returned. This is the same as not providing a type argument.
2	A list of all selected ranges is returned. The current cursor location is considered a selected range.
3	One of any marked element.
4	Only the range that was selected first is returned.
5	All selected rows and/or columns are returned.
6	One descriptor.
7	A list of all selected columns is returned.
8	One column.

- 9 The cell address of the cell that currently contains the cursor is returned.
- 14 Returns the width setting for the currently selected column.
- 21 Returns the currency symbol set for Spreadsheets.
- 25 One row.
- 27 List of rows.

Description This macro is obsolete. Please use [SS_SELECTED@](#) instead.

Returns information on selected cells. Any ranges returned are represented using range address notation. For example, if ``C3, B8, D5..E10" is returned, it indicates that cells C3, B8, and all cells in the range D5 through E10 are selected.

- See also** [SS_SELECT_ALL@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_CLEAR@](#)
[SS_SELECT_DOWN@](#)
[SS_SELECT_LEFT@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_RIGHT@](#)
[SS_SELECT_UP@](#)

SS_SELECT_ALL@

Selects all cells containing data

Format SS_SELECT_ALL@()

Method [this.select_all@](#)

Description Selects all cells between cell A1 and the last cell containing data. SS_SELECT_ALL@ is called by the Edit ® Select ® All menu option.

- See also** [SS_SELECT@](#)
[SS_SELECT_ALL_SHEETS@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_CLEAR@](#)
[SS_SELECT_DOWN@](#)

[SS_SELECT_LEFT@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_RIGHT@](#)
[SS_SELECT_UP@](#)

SS_SELECT_ALL_SHEETS@

Selects all cells containing data on all sheets

Format SS_SELECT_ALL_SHEETS@()

Method [this.select_all_sheets@](#)

Description Selects all cells between cell A1 and the last cell containing data on all sheets. SS_SELECT_ALL_SHEETS@ is called by the Edit ® Select ® All Sheets menu option.

See also [SS_SELECT@](#)
[SS_SELECT_ALL@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_CLEAR@](#)
[SS_SELECT_DOWN@](#)
[SS_SELECT_LEFT@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_RIGHT@](#)
[SS_SELECT_UP@](#)

SS_SELECT_ARRAY_FORMULA@

Selects all of an array's output

Format SS_SELECT_ARRAY_FORMULA@()

Method [this.select_array_formula@](#)

Description If the cell cursor is within the output of an array function, selects the entire output.

See also [SS_SELECT@](#)
[SS_SELECT_ALL@](#)
[SS_SELECT_CLEAR@](#)
[SS_SELECT_DOWN@](#)
[SS_SELECT_LEFT@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_RIGHT@](#)
[SS_SELECT_UP@](#)

SS_SELECT_BOTTOM_SECTION@

Extends selection downward

Format SS_SELECT_BOTTOM_SECTION@()

Method [this.select_bottom_section@](#)

Description Extends the selection downward in the spreadsheet. The distance of the extension depends on the contents of the currently selected cells.

If the currently selected cells are empty, the range is extended downward to include the first set of cells that contain a value.

If the currently selected cells contain a value, and any cell immediately below the selection contains a value, the selection is extended downward a set of cells that contain no value.

If the currently selected cells contain a value, and all cells immediately below the selection contain no value, the selection is extended until it includes the first set of cells that contain a value. If there are no cells below the selection that contain a value, the selection is not extended.

See also [SS_SELECT_LEFT_SECTION@](#)
[SS_SELECT_RIGHT_SECTION@](#)
[SS_SELECT_TOP_SECTION@](#)

SS_SELECT_CLEAR@

Deselects all selected cells

Format SS_SELECT_CLEAR@()

Method [this.select_clear@](#)

Description Deselects all selected cells in the current spreadsheet. All selections are cleared by a call to SS_SELECT_CLEAR@.

See also [SS_SELECT@](#)
[SS_SELECT_ALL@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_DOWN@](#)
[SS_SELECT_LEFT@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_RIGHT@](#)
[SS_SELECT_UP@](#)

SS_SELECT_DOWN@

Selects the current cell and the cell below it

Format SS_SELECT_DOWN@()

Method [this.select_down@](#)

Description Selects the current spreadsheet cell and the cell below it in a spreadsheet. SS_SELECT_DOWN@ is called by the Keys ® Range select down menu option.

See also [SS_SELECT@](#)
[SS_SELECT_ALL@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_CLEAR@](#)

[SS_SELECT_LEFT@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_RIGHT@](#)
[SS_SELECT_UP@](#)

SS_SELECT_LEFT@

Selects the current cell and the cell to its left

Format SS_SELECT_LEFT@()

Method [this.select_left@](#)

Description Selects the current spreadsheet cell and the cell to its left in a spreadsheet. SS_SELECT_LEFT@ is called by the Keys ® Range select left menu option.

See also [SS_SELECT@](#)
[SS_SELECT_ALL@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_CLEAR@](#)
[SS_SELECT_DOWN@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_RIGHT@](#)
[SS_SELECT_UP@](#)

SS_SELECT_LEFT_SECTION@

Extends selection leftward

Format SS_SELECT_LEFT_SECTION@()

Method [this.select_left_section@](#)

Description Extends the selection leftward in the spreadsheet. The distance of the extension depends on the contents of the currently selected cells.

If the currently selected cells are empty, the range is extended leftward to include the first set of cells that contain a value.

If the currently selected cells contain a value, and any cell immediately to the left of the selection contains a value, the selection is extended leftward until it encounters the edge of the spreadsheet or a set of cells that contain no value.

If the currently selected cells contain a value, and all cells immediately to the left of the selection contain no value, the selection is extended until it includes the first set of cells that contain a value. If there are no cells to the left of the selection that contain a value, the selection is not extended.

SS_SELECTED@

Returns the addresses or other information for selected cells

Format info = SS_SELECTED@([type[, pos]])

Method info = this.selected@([type[, pos]])

Arguments	type	A string that specifies the criteria for the cell addresses or other information that will be returned. If you do not include type, the cell address of all selected cells is returned. type can be any of the following:
	1	All selected cells are returned. This is the same as not providing a type argument.
	2	A list of all selected ranges is returned. The current cursor location is considered a selected range.
	3	One of any marked element.
	4	Only the range that was selected first is returned.
	5	All selected rows and/or columns are returned.
	6	One descriptor.
	7	A list of all selected columns is returned.
	8	One column.
	9	The cell address of the cell that currently contains the cursor is returned.
	14	Returns the width setting for the currently selected column.
	21	Returns the currency symbol set for Spreadsheets.
	25	One row.
	27	List of rows.

Description Returns information on selected cells. Any ranges returned are represented using range address notation. For example, if ``C3, B8, D5..E10" is returned, it indicates that cells C3, B8, and all cells in the range D5 through E10 are selected.

See also [SS_SELECT_ALL@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_CLEAR@](#)
[SS_SELECT_DOWN@](#)
[SS_SELECT_LEFT@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_RIGHT@](#)
[SS_SELECT_UP@](#)

SS_SELECT_MATERIAL@

Selects the specified spreadsheet cells

Format SS_SELECT_MATERIAL@(cells)

Method [this.select_material@\(cells\)](#)

Arguments cells A string indicating the cells to select, as follows:

- An individual cell reference.
- A range of cells as specified by an existing range name or by the beginning and ending cell range references.
- A column of cells as specified by a column letter.
- A row of cells as specified by the row number. The number must be specified as a string, such as ``3".
- A non-contiguous group of cells. Each cell is specified by its cell reference and the cell references are separated by commas. For example, ``C1, D5, F12, F19".

Description Selects a cell, range of cells, group of non-contiguous cells, a row of cells, or a column of cells, depending on the value of the cells variable. Hidden cells can be selected using SS_SELECT_MATERIAL@. They are displayed as selected when revealed.

See also [SS_SELECT@](#)
[SS_SELECT_ALL@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_CLEAR@](#)
[SS_SELECT_DOWN@](#)
[SS_SELECT_LEFT@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_RIGHT@](#)
[SS_SELECT_UP@](#)

SS_SELECT_OBJECT@

Selects an object by name

Format SS_SELECT_OBJECT@(name)

Method [this.select_object@](#)(name)

Arguments name The name of the object in the spreadsheets document.

Description Selects an object by name in the spreadsheets document. Any object selected prior to using SS_SELECT_OBJECT@ remain selected.

SS_SELECT_PAGEBREAKS@

Selects all manual page breaks

Format SS_SELECT_PAGEBREAKS@()

Method [this.select_pagebreaks@](#)

Description Selects all rows and columns in the current spreadsheet for which a manual page break has been set. The page will break before the selected rows or columns. Nothing is selected if no manual page breaks were set.

See also [SS_SELECT@](#)
[SS_SELECT_ALL@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_CLEAR@](#)
[SS_SELECT_DOWN@](#)
[SS_SELECT_LEFT@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_RIGHT@](#)
[SS_SELECT_UP@](#)

SS_SELECT_PROTECTED@

Selects all protected cells

Format SS_SELECT_PROTECTED@()

Method [this.select_protected@](#)

Description Selects all cells in the spreadsheet that have been set to protected. This macro is bound to Edit ® Select ® Protected Cells.

See also [SS_SELECT@](#)
[SS_SELECT_ALL@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_CLEAR@](#)
[SS_SELECT_DOWN@](#)
[SS_SELECT_LEFT@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_RIGHT@](#)
[SS_SELECT_UP@](#)

SS_SELECT_RIGHT@

Selects the current cell and the cell to its right

Format SS_SELECT_RIGHT@()

Method [this.select_right@](#)

Description Selects the current spreadsheet cell and the cell to its left in a spreadsheet. SS_SELECT_RIGHT@ is called by the Keys® Range select left menu option.

See also [SS_SELECT@](#)
[SS_SELECT_ALL@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_CLEAR@](#)
[SS_SELECT_DOWN@](#)
[SS_SELECT_LEFT@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_UP@](#)

SS_SELECT_RIGHT_SECTION@

Extends selection to the right

Format SS_SELECT_RIGHT_SECTION@()

Method [this.select_right_section@](#)

Description Extends the selection to the right in the spreadsheet. The distance of the extension depends on the contents of the currently selected cells.

If the currently selected cells are empty, the range is extended rightward to include the first set of cells that contain a value. If no cells to the right of the selection contain a value, the selection is not extended.

If the currently selected cells contain a value, and any cell immediately to the right of the selection contains a value, the selection is extended rightward to the first set of cells that contain no value.

If the currently selected cells contain a value, and all cells immediately to the right of the selection contain no value, the selection is extended until it includes the first set of cells that contain a value. If there are no cells below the selection that contain a value, the selection is not extended.

See also [SS_SELECT_BOTTOM_SECTION@](#)
[SS_SELECT_LEFT_SECTION@](#)
[SS_SELECT_TOP_SECTION@](#)

SS_SELECT_TOP_SECTION@

Extends selection upward

Format SS_SELECT_TOP_SECTION@()

Method [this.select_top_section@](#)

Description Extends the selection upward in the spreadsheet. The distance of the extension depends on the contents of the currently selected cells.

If the currently selected cells are empty, the range is extended upward to include the first set of cells that contain a value. If no cells above the selection contain a value, the selection is not extended.

If the currently selected cells contain a value, and any cell immediately above the selection contains a value, the selection is extended upward to the first set of cells that contain no value, or to the edge of the spreadsheet.

If the currently selected cells contain a value, and all cells immediately above the selection contain no value, the selection is extended until it includes the first set of cells above the selection that contain a value. If there are no cells above the selection that contain a value, the selection is not extended.

See also [SS_SELECT_BOTTOM_SECTION@](#)
[SS_SELECT_LEFT_SECTION@](#)
[SS_SELECT_RIGHT_SECTION@](#)

SS_SELECT_UP@

Selects the current cell and the cell immediately above it

Format SS_SELECT_UP@()

Method [this.select_up@](#)

Description Selects the current spreadsheet cell and the cell above it in a spreadsheet. SS_SELECT_UP@ is called by the Keys ® Range select up menu option.

See also [SS_SELECT@](#)
[SS_SELECT_ALL@](#)
[SS_SELECT_ARRAY_FORMULA@](#)
[SS_SELECT_CLEAR@](#)
[SS_SELECT_DOWN@](#)
[SS_SELECT_LEFT@](#)
[SS_SELECT_MATERIAL@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_SELECT_PROTECTED@](#)
[SS_SELECT_RIGHT@](#)

SS_SETCR_VALUE@

Sets the "RETURN" cursor movement direction

Format SS_SETCR_VALUE@(direction)

Method [this.setcr_value@\(direction\)](#)

Arguments

direction	A number that indicates the direction in which to move the cursor when RETURN is pressed. direction can be one of the following:
0	cursor moves down one cell
1	cursor moves one cell to right
3	cursor does not move from current cell

Description Indicates the direction the spreadsheet cursor should move when RETURN is pressed. The cursor movement you specify applies to all Spreadsheets windows.

See also [SS_RETURN_KEY@](#)

SS_SETPTR_VALUE@

Displays the mouse pointer using the symbol specified

Format SS_SETPTR_VALUE@(symbol)

Method [this.setptr_value@\(symbol\)](#)

Arguments symbol A string that is the name of the mouse pointer symbol that you want to display. The macro **GET CURSOR LIST@** returns an array of mouse pointer names.

Description Specifies which mouse pointer symbol is displayed while the pointer is in the work area of a spreadsheet. The mouse pointer specified applies to all Spreadsheets windows.

SS_SET_ASCII_FORMATTED@

Determines whether spreadsheets are exported formatted or unformatted

Format SS_SET_ASCII_FORMATTED@(formatted)

Method [this.set_ascii_formatted@\(formatted\)](#)

Arguments formatted Determines whether to export formatted or unformatted Spreadsheets:
0 Strip spreadsheets of their formats when exporting them.
1 Preserve format when exporting spreadsheets.

Description Turns on and off the switch that determines whether spreadsheets are formatted or not when they are exported.

SS_SET_ASCII_FORMATTED@ is called by Q ® Spreadsheet Preferences.

SS_SET_ASCII_QUOTE_LABELS@

Indicates if quotes are used in an ASCII import

Format SS_SET_ASCII_QUOTE_LABELS@(quoteLabelsFlag)

Method [this.set_ascii_quote_labels@\(quoteLabelsFlag\)](#)

Arguments quoteLabelsFlag
A Boolean value which if set to TRUE indicates that quotation marks will be used.

Description Places quotation marks around the labels in imported ASCII files; this flag is placed in the ax_prof4 file.

See also [SS_IMPORT_ASCII@](#)
[SS_SET_ASCII_TRIM_WHITESPACE@](#)

SS_SET_ASCII_TRIM_WHITESPACE@

Indicates if whitespace is removed when information is imported

Format SS_SET_ASCII_TRIM_WHITESPACE@(flag)

Method [this.set_ascii_trim_whitespace@\(flag\)](#)

Arguments flag A Boolean value where TRUE means whitespace is removed.

Description Indicates if whitespace (spaces and tabs) are removed when ASCII information is imported into a Spreadsheets document.

See also [SS_IMPORT_ASCII@](#)
[SS_SET_ASCII_QUOTE_LABELS@](#)

SS_SET_AUTO_RESIZE_ROWS@

Toggles "auto-resize row"

Format SS_SET_AUTO_RESIZE_ROWS@ (autoResizeRowsFlag)

Method [this.set_auto_resize_rows@ \(autoResizeRowsFlag\)](#)

Arguments autoResizeRowsFlag
A Boolean value which if set to TRUE indicates that rows are automatically resized.

Description Turns on or off the auto-resize row feature. Using this macro is the same as turning on the * ® Spreadsheets Preferences Auto Resize Rows option.

SS_SET_AUTOBACKUP_PROFILE@

Sets the time interval at which spreadsheet files are backed up

Format SS_SET_AUTOBACKUP_PROFILE@(interval)

Method [this.set_autobackup_profile@\(interval\)](#)

Arguments interval The backup interval.

Description Creates a backup copy of a Spreadsheets document at specific intervals. The copy is saved in the directory named in the [Auto-save directory pathname](#) option.

If the Spreadsheets document has not been saved or backed up, and has been edited within the specified amount of time, a backup copy of the current Spreadsheets document is automatically saved. The backup copy is saved, with the current document name without affecting the current document's status.

The interval is measured from the most recent edit since the last save or backup. By default this is set to 0 (never) which does not do an automatic backup.

Set to 0 to prevent automatic backup.

Enter a whole number representing the interval in minutes, since the most recent edit, at which to create a backup copy.

SS_SET_BORDERS@

Sets border attributes

Format SS_SET_BORDERS@(selection, format borders_info_ attributes)

Method [this.set_borders@\(selection, format borders_info_ attributes\)](#)

Arguments selection A string indicating a cell address or a spreadsheet range.
attributes An array of cell border attributes. The FORMAT template for borders_info_. The header file containing the FORMAT template is spsheet_.am. The elements within attributes are described below.

Description Sets the cell border attributes for the given selection. SS_SET_BORDERS@ is called by the Style ® Borders menu option.

The definition of the borders_info_ attributes variable is as follows:

format borders_info_

```

format ss_line_attrs outline,
format ss_line_attrs top,
    Same values as outline
format ss_line_attrs bot,
    Same values as outline
format ss_line_attrs left,
    Same values as outline
format ss_line_attrs right,
format ss_shade attrs shading

```

The definition of `ss_line_attrs` is as follows:

```

format ss_line_attrs
    style,  -1  mixed
            0  off
            1  thin
            2  medium
            3  thick
            4  dashed
            5  double
    color   The color's string name

```

The definition of `ss_shade_attrs` is as follows:

```

format ss_shade_attrs
    style,  0 to 19: These numbers (which represent the shading patterns) can be
            derived from the shade palette, reading left to right, top to bottom.
    fgcolor, The color's string name
    bgcolor  The color's string name

```

See also [SS GET BORDERS@](#).

SS_SET_CALC_OPTIONS@

Sets the 'calc' options

Format `SS_SET_CALC_OPTIONS@(format ss_calc_options@ calcOptions)`

Method `this.set_calc_options@(format ss_calc_options@ calcOptions)`

Arguments `calcOptions` The 'calc' options being set.

Description Sets the recalculation options to `calcOptions`. The format of this variable is `ss_calc_options@`; its definition is:

```

format ss_calc_options@
    mode,          '0: manual calculation

```

	'1: auto calculation
	'2: interval calculation
style,	'0: Normal recalculation
	'1: Row recalculation
	'2: Column recalculation
iteration_count,	'Number between 1 and 10 indicating the number of times a value is calculated if multiple dependencies exist
calc_interval,	'Time in seconds between recalculations
auto_chart,	'Boolean: TRUE indicates charts are updated when dependent values change
calc_background,	'Boolean: TRUE turns on background calculation
calc_on_display,	'Boolean: TRUE keeps the spreadsheet display current
calc_rtinsert_on_display,	'Boolean: calculates any obsolete formulas containing RTINSERT functions
calc_only_obsolete_cells,	'Boolean
type_conversion	'Boolean: type conversion for text to numeric or vice-versa

SS_SET_CALLBACK@

Installs a ``selection" callback function

Format SS_SET_CALLBACK@(macro)

Method [this.set_callback@\(macro\)](#)

Arguments macro The macro called when a selection is made. This ``callback macro" is disabled with [SS_UNSET_CALLBACK@](#).

Description Installs a callback function that allows you to automatically invoke a particular macro ¾ the ``callback macro" ¾ every time you make a selection in the grid area of a Spreadsheet. The callback macro can apply to the current cell or a range of cells.

The callback macro must begin with a [DELAY@](#). We also recommend that you run and test the callback macro alone before calling it using [SS_SET_CALLBACK@](#).

See also [SS_UNSET_CALLBACK@](#)
[SS_SET_CR_CALLBACK@](#)
[SS_UNSET_CR_CALLBACK@](#)
[SS_SET_DBL_CLICK_CALLBACK@](#)
[SS_UNSET_DBL_CLICK_CALLBACK@](#)
[SS_SET_LOAD_CELL_CALLBACK@](#)

[SS_UNSET_LOAD_CELL_CALLBACK@](#)

SS_SET_CELL_ATTR_INFO@

Sets the format attributes for a range of cells

Format SS_SET_CELL_ATTR_INFO@(range, format ss_cell_attr_info@ attr_info)

Arguments range A string containing a range of cells, such as A:B4..A:C7.
attr_info A formatted array. The ELF format ss_cell_attr_info@ is described in the file [spsheet.am](#).

Description Applies formatting to a range of cells in a spreadsheet.

SS_SET_CELL_FONT_SIZE@

Changes the font size

Format SS_SET_CELL_FONT_SIZE@(size)

Method [this.set_cell_font_size@](#)(size)

Arguments size The point size of a font. size must be one of the following values: 6, 8, 10, 12, 14, 18, 24, or 36.

Description Sets the font size for the current selection to size.

See also [SS_GET_FONT_SIZES@](#)

SS_SET_CHART_GROUP@

Sets the chart's data groups

Format SS_SET_CHART_GROUP@(chartName, group, data, noUpdateFlag)

Method [this.set_chart_group@](#)(chartName, group, data, noUpdateFlag)

Arguments chartName The name of a chart.

group The identifier for a group. A default group name created by Applixware Spreadsheets is in the following form:

```

data 0
data 1
data 2

```

 However, the value following data can be any user-defined string.

data The data for the group.

noUpdateFlag A Boolean value, which if set to FALSE tells Applixware Spreadsheets that the chart should be updated after the group's data is interpreted.

Description Defines the data associated with a group.

See also [SS SET CHART GROUP PROFILE@](#)

SS_SET_CHART_GROUP_PROFILE@

Sets a profile indicating how ranges are grouped

Format SS_SET_CHART_GROUP_PROFILE@(groupOrder)

Method [this.set_chart_group_profile@\(groupOrder\)](#)

Arguments groupOrder One of the following values:

```

Column
Row

```

Description Sets the user's profile to indicate if ranges are grouped by column or by row.

See also [SS SET CHART GROUP@](#)

SS_SET_CHART_NAVIGATE@

Sets the chart navigate state

Format SS_SET_CHART_NAVIGATE@(chartNavigate)

Method [this.set_chart_navigate@\(chartNavigate\)](#)

Arguments chartNavigate One of the following:

- 0 Off (use defaults)
- 1 One-step
- 2 Step-by-step
- 3 Template

Description Sets the chart navigate state, as follows:

- 0 Use the default settings when you create a chart by clicking on the Create Chart icon in the *ExpressLine*.
- 1 Preview and create a chart in the Chart Data dialog box when you click on the Create Chart icon in the *ExpressLine*.
- 2 Create a chart in stages which include Chart Data, Chart Types, and Chart Labels when you click on the Create Chart icon in the *ExpressLine*.
- 3 Use the attributes of a pre-existing chart as a template.

See also [SS SET CHART NAVIGATE PROFILE@](#)

SS SET CHART NAVIGATE PROFILE@

Sets the chart navigate profile

Format SS_SET_CHART_NAVIGATE_PROFILE@ (chartNavigate)

Method [this.set_chart_navigate_profile@](#) (chartNavigate)

Arguments chartNavigate

One of the following:

- 0 Off (use defaults)
- 1 One-step
- 2 Step-by-step
- 3 Template

Description Sets the chart navigate state profile so that one of the following actions will occur when creating a chart:

- 0 Use the default settings when you create a chart by clicking on the Create Chart icon in the *ExpressLine*.
- 1 Preview and create a chart in the Chart Data dialog box when you click on the Create Chart icon in the *ExpressLine*.
- 2 Create a chart in stages which include Chart Data, Chart Types, and Chart Labels when you click on the Create Chart icon in the *ExpressLine*.
- 3 Use the attributes of a pre-existing chart as a template.

See also [SS_SET_CHART_NAVIGATE@](#)

SS_SET_COL_WIDTH@

Sets column width

Format SS_SET_COL_WIDTH@(cols, width, bestFitFlag)

Method [this.set_col_width@](#)(cols, width, bestFitFlag)

Arguments

cols	The columns whose width is being set.
width	The width to which the columns will be set. This argument indicates the width in characters of the column.
bestFitFlag	A Boolean value which if set to TRUE indicates that Spreadsheets will manipulate the width of cells that cannot be set to width. If FALSE, cells are set to width.

Description Sets the column width for the indicated columns. If your width is too large or too small and you have not set bestFitFlag to TRUE, an error is thrown.

See also [SS_GET_COL_WIDTH@](#)

SS_SET_COPIED_ATTRS_PROFILE@

Sets "copy attribute" preference

Format SS_SET_COPIED_ATTRS_PROFILE@ (useCopiedAttrsFlag)

Method [this.set_copied_attrs_profile@](#) (useCopiedAttrsFlag)

Arguments

useCopiedAttrsFlag	A Boolean value which if set to TRUE indicates that attributes will be copied.
--------------------	--

Description Sets a Spreadsheets preference so that cell attributes are copied along with cell formulas and values.

SS_SET_CR_CALLBACK@

Installs the RETURN callback macro

Format SS_SET_CR_CALLBACK@(macro)

Method [this.set_cr_callback@\(macro\)](#)

Arguments macro The macro to be invoked immediately upon pressing the RETURN key.

Description Installs a callback function that allows you to automatically invoke a particular macro ¾ the ``callback macro" ¾ every time you press the RETURN key in a cell. SS_SET_CR_CALLBACK@ is disabled using the [SS_UNSET_CR_CALLBACK@](#) macro.

The callback macro must begin with a [DELAY@](#). We also recommend that you run and test the callback macro alone before calling it using SS_SET_CR_CALLBACK@.

See also [SS_SET_CALLBACK@](#)
[SS_SET_DBL_CLICK_CALLBACK@](#)
[SS_UNSET_CALLBACK@](#)
[SS_UNSET_DBL_CLICK_CALLBACK@](#)
[SS_UNSET_LOAD_CELL_CALLBACK@](#)

SS_SET_CURRENCY@

Sets how currency values are displayed

Format SS_SET_CURRENCY@(symbol, europeFlag, trailingFlag)

Method [this.set_currency@\(symbol, europeFlag, trailingFlag\)](#)

Arguments symbol A string indicating the currency symbol.
europeFlag A Boolean value which if set to TRUE indicates that currency information is to be displayed in European format.
trailingFlag A Boolean value which if set to TRUE indicates that the currency symbol is displayed after the currency's numeric value.

Description Defines how currency values are displayed. The numeric value of the currency is not altered.

See Also [SS_GET_CURRENCY@](#)

SS_SET_CURRENCY_POS_PROFILE@

Sets the default currency position to left or right of value

Format SS_SET_CURRENCY_POS_PROFILE@(position)

Method [this.set_currency_pos_profile@\(position\)](#)

Arguments position A Boolean value where TRUE means to place the currency symbol to the left of the number.

Description Specifies where the currency symbol appears in relation to the currency number.

See also the [Decimal separator character](#) and [Set the default currency symbol](#) preference options.

SS_SET_CURRENCY_SYMB_PROFILE@

Defines the currency symbol

Format SS_SET_CURRENCY_SYMB_PROFILE@(symbol)

Method [this.set_currency_symb_profile@\(symbol\)](#)

Arguments symbol The currency symbol to be used.

Description Specifies the default symbol for money. The currency symbol automatically appears in the Spreadsheet when the financial or other currency-type number format is used.

Enter the ASCII value for the currency symbol you want to use. By default \$ is used. To change that, enter the ASCII value for the symbol you want to use:

Cent=162

Sterling=163

Yen=165

See also the [Decimal separator character](#) and [Place the currency symbol left of the number](#) preference options.

SS_SET_CURSOR_PROFILE@

Sets the mouse pointer image and cursor movement attributes

Format SS_SET_CURSOR_PROFILE@(pointer, cursorMove)

Method [this.set_cursor_profile@](#)(pointer, cursorMove)

Arguments

pointer	A string indicating the symbol to use for the mouse pointer. For a list of possible symbol values, see SET LOOK FEEL VALUES@ .
cursorMove	One of the following values: 0 Cursor moves down one cell 1 Cursor moves to the right one cell 2 Cursor remains in the current cell

Description Indicates how the cursor is moved when RETURN is pressed.
The default pointer is a cross. The default cursorMove style when RETURN is pressed is to move down one cell.

SS_SET_DATABASE@

Sets or changes a database entity within the Spreadsheet

Format SS_SET_DATABASE@(oldName, newName, dbRange, critRange[, extRange])

Method [this.set_database@](#)(oldName, newName, dbRange, critRange[, extRange])

Arguments

oldName	The current name of the on-sheet database.
newName	The new name of the on-sheet database.
dbRange	The range in which the database is stored.
critRange	The range which contains the criterion information.
extRange	The range into which the extracted information is written.

Description Creates a new onsheet database or changes attributes of an existing database.

See also [SS_GET_DB_INFO@](#)

SS_SET_DBL_CLICK_CALLBACK@

Installs a "double click" callback

Format SS_SET_DBL_CLICK_CALLBACK@(macro)

Method [this.set_dbl_click_callback@\(macro\)](#)

Arguments macro The macro to be called whenever you double-click in a Spreadsheet cell.

Description Installs a callback function that allows you to automatically invoke a particular macro ^¾ the "callback macro" ^¾ every time you double-click (with a mouse) in a cell.

SS_SET_DBL_CLICK_CALLBACK@ is disabled using the [SS_UNSET_DBL_CLICK_CALLBACK@](#) macro.

The callback macro must begin with a [DELAY@](#). We also recommend that you run and test the callback macro alone before calling it using the SS_SET_DBL_CLICK_CALLBACK@ macro. If a cell contains a macro, that macro takes precedence over the double-click callback.

See also [SS_UNSET_DBL_CLICK_CALLBACK@](#)

[SS_SET_CALLBACK@](#)

[SS_UNSET_CALLBACK@](#)

[SS_SET_CR_CALLBACK@](#)

[SS_UNSET_CR_CALLBACK@](#)

[SS_SET_LOAD_CELL_CALLBACK@](#)

[SS_UNSET_LOAD_CELL_CALLBACK@](#)

SS_SET_DELIMITER_PROFILE@

Indicates the ASCII character separator

Format SS_SET_DELIMITER_PROFILE@(delimiter)

Method [this.set_delimiter_profile@\(delimiter\)](#)

Arguments delimiter The ASCII character to be interpreted as the delimiter which separates cells in documents that are imported to, or exported from, Spreadsheets. The most common values for delimiter include:

- 0 (NULL, the default): Word spaces and tabs are treated as the delimiter. Carriage return is treated as line break.
- 9 A horizontal tab is treated as a delimiter. Carriage return is treated as line break.
- 10 A line feed is treated as a delimiter.
- 12 A form feed is treated as a delimiter.
- 13 A carriage return is treated as a delimiter.
- 35 A pound sign (#) is treated as a delimiter. Carriage return is treated as line break.
- 36 A dollar sign (\$) is treated as a delimiter. Carriage return is treated as line break.
- 37 A percent sign (%) is treated as a delimiter. Carriage return is treated as line break.
- 44 A comma is treated as a delimiter. Carriage return is treated as line break.
- 127 A word space is ignored. Carriage return is treated as line break.

Description Indicates the delimiter that is used to separate cells in documents. `SS_SET_DELIMITER_PROFILE@` is called by the ASCII Delimiter entry box in the *® Spreadsheets Preferences menu option.

SS_SET_DOC_ATTR@

Sets document attributes

Format `SS_SET_DOC_ATTR@(numStyle, numPrec, numAlign, textAlign, boldFlag, italicFlag, underline, faceName, color, displaySize, wrapText[, defColumnWidth])`

Method `this.set_doc_attr@(numStyle, numPrec, numAlign, textAlign, boldFlag, italicFlag, underline, faceName, color, displaySize, wrapText[, defColumnWidth])`

Arguments numStyle The default numeric style of all cells, as follows:

- 0 Unstyled
- 1 Boolean
- 2 General
- 3 Fixed
- 4 Scientific
- 5 Money
- 6 Comma
- 7 Percent

	8	Date
	9	Graph
	10	Time
	11	Default
numPrec	The default numeric precision of displayed numbers. This is a value from 0 to 9.	
numAlign	The alignment of numeric information within a cell, as follows:	
	0	Left
	1	Right
	2	Center
	3	Repeat
textAlign	The alignment of label information within a cell, as follows:	
	0	Left
	1	Right
	2	Center
boldFlag	A Boolean value which if set to TRUE indicates that the default display of information is in bold.	
italicFlag	A Boolean value which if set to TRUE indicates that the default display of information is in italics.	
underline	A value which indicates if underlines are drawn, and if they are, if they are single or double underlines, as follows:	
	0	None
	1	Single
	2	Double
faceName	The string name of the default font for the spreadsheet.	
color	The name of the color in which information is displayed.	
displaySize	The default point size of displayed information.	
wrapText	A Boolean value which if set to TRUE indicates that text will be wrapped within cells rather than extended to print over cells to the right.	
defColWidth	The default column width. This width is measured in terms of the number of characters that can be entered. If this argument is omitted, the default column width is 10 characters wide.	

Description Sets the default spreadsheet display attributes, as indicated above.

See also [**SS GET DOC ATTR@**](#).

SS_SET_DRAG_COPYPASTE@

Allows you to move a range by dragging

Format SS_SET_DRAG_COPYPASTE@(dragFlag)

Method [this.set_drag_copypaste@\(dragFlag\)](#)

Arguments dragFlag A Boolean value which if set to TRUE indicates that you can move the selected cell range by dragging it with a mouse. When the mouse pointer is placed over one of the edges of the selected range, it changes into a fleur



CURSOR Fleur Cursor

SS_SET_DRAG_COPYPASTE_PROFILE@

Turns on or off the drag/drop profile

Format SS_SET_DRAG_COPYPASTE_PROFILE@(dragCopyPasteFlag)

Method [this.set_drag_copypaste_profile@\(dragCopyPasteFlag\)](#)

Arguments dragCopyPasteFlag A Boolean value which if set to TRUE indicates that you can move a selected range by dragging it with the mouse.

Description Sets a profile entry that allows you to move a range by dragging instead of using the copy and paste commands.

SS_SET_EDIT_FACE_PROFILE@

Sets the default typeface style

Format SS_SET_EDIT_FACE_PROFILE@(typeface)

Method [this.set_edit_face_profile@\(typeface\)](#)

Arguments typeface The name of the typeface. The values of typeface can be one of the following strings, depending on the font families set in your [Printing Preferences](#).

[PCL5 font family](#)

[PostScript font family](#)

Description Sets the default type face to the passed string in typeface.

SS_SET_EDIT_SIZE_PROFILE@

Defines the default point size

Format SS_SET_EDIT_SIZE_PROFILE@(size)

Method [this.set_edit_size_profile@\(size\)](#)

SS_SET_GRID_LINES@

Sets grid lines

Format SS_SET_GRID_LINES@()

Method [this.set_grid_lines@](#)

Description When executed, spreadsheet grid lines will be displayed as dotted, solid, or none (not displayed), depending on the Grid Lines setting in the [Spreadsheets Preferences](#) dialog box.

See also [SS_GRID_LINES@](#)

SS_SET_GRID_PROFILE@

Sets the grid display style

Format SS_SET_GRID_PROFILE@(gridStyle)

Method [this.set_grid_profile@\(gridStyle\)](#)

Arguments gridStyle A number indicates the style in which to display grids in the Spreadsheets window. gridStyle can be:

0	none
1	dotted grid lines
2	solid grid lines

Description Sets the grid display setting for the current Spreadsheets document. The default grid_style is dotted.

SS_SET_HDRFTR@

Sets the headers and footers within the Spreadsheets document

Format SS_SET_HDRFTR@(format hdrftr_info data)

Method [this.set_hdrftr@](#)(format hdrftr_info data)

Arguments data The data defining a Spreadsheet's headers and footers.

Description Sets the information that will appear in the printed Spreadsheet's headers and footers. For a description of the hdrftr_info format, see [SS_GET_HDRFTR@](#).

SS_SET_HOOK@

Installs a macro called when the Spreadsheets file is opened

Format SS_SET_HOOK@(macroName)

Method [this.set_hook@](#)(macroName)

Arguments macroName The name of the macro. If macroName is not loaded when this macro executes, ELF will load, compile, and execute a file with the same name (after a .am extension is appended) as macroName that exists in your default path.

Description Embeds a macro within a Spreadsheets document that is run whenever the Spreadsheets document is run. To unset the SS_SET_HOOK@ ``hook" in a Spreadsheets document, run SS_SET_HOOK@ without any arguments.

See also [SS_GET_HOOK@](#)

SS_SET_INITMACRO_PROFILE@

Names the macro that is run when a Spreadsheet file is opened

Format SS_SET_INITMACRO_PROFILE@(macroName)

Method [this.set_initmacro_profile@\(macroName\)](#)

Arguments macroName The name of the macro that will be run automatically when any Spreadsheets file or window is opened.

SS_SET_LINKUPDATE_PROFILE@

Indicates the seconds between checks to see if links need updating

Format SS_SET_LINKUPDATE_PROFILE@(interval)

Method [this.set_linkupdate_profile@\(interval\)](#)

Arguments interval The number of seconds between checks.

Description Sets the interval in seconds between checks to see if non-Applixware files linked to the Spreadsheets document have changed.

The default is 10 seconds.

SS_SET_LOAD_CELL_CALLBACK@

Installs a "new value" callback

Format SS_SET_LOAD_CELL_CALLBACK@(macro)

Method [this.set_load_cell_callback@\(macro\)](#)

Arguments macro The macro to be invoked immediately upon loading a new value into a Spreadsheet cell.

Description Installs a callback function that allows you to automatically invoke a particular macro, (the "callback macro") every time you load a new value into a cell. SS_SET_LOAD_CELL_CALLBACK@ can help you compare new and previous entries and thereby accept or forbid new entries based on their appropriateness.

The callback macro must begin with a **DELAY@**. We also recommend that you run and test the callback macro alone before calling it using the `SS_SET_LOAD_CELL_CALLBACK@` macro.

Use **SS_UNSET_LOAD_CELL_CALLBACK@** to disable this macro.

See also **SS_SET_CALLBACK@**
SS_UNSET_CALLBACK@
SS_SET_CR_CALLBACK@
SS_UNSET_CR_CALLBACK@
SS_SET_DBL_CLICK_CALLBACK@
SS_UNSET_DBL_CLICK_CALLBACK@

SS_SET_MOVE_PREFERENCE@

Sets how ranges are adjusted during a move

Format `SS_SET_MOVE_PREFERENCE@(value)`

Method `this.set_move_preference@(value)`

Arguments value A value determining the method used to adjust range references during a move. The values are:

0	Use range corners
1	Use range sides

See also **SS_SET_MOVE_PREFERENCE_PROF@**

SS_SET_MOVE_PREFERENCE_PROF@

Sets the adjust range references during move preference

Format `SS_SET_MOVE_PREFERENCE_PROF@(value)`

Method `this.set_move_preference_prof@(value)`

Arguments value A value determining the method used to adjust range references during a move. The values are:

0	Use range corners
1	Use range sides

Description Sets the preference for determining the method used to adjust range references during a move.

See also [SS_SET_MOVE_PREFERENCE@](#) and the [Adjust Range References during Move Using](#) preference option.

SS_SET_OBJ_INFO@

Defines an object's properties

Format SS_SET_OBJ_INFO@(name, format ss_obj_info@ obj)

Method [this.set_obj_info@](#)(name, format ss_obj_info@ obj)

Arguments

name	The name of an object embedded within the spreadsheet.
obj	The information being defined for the object.

Description Defines the properties of object name. These properties are formatted in an ss_obj_info@ format whose definition is as follows:

```
format ss_obj_info@
    name,          'string
    type,          'integer
    property,      'Boolean
    hidden,        'Boolean
    locked,        'Boolean
    extlink,       'Boolean
    print,         'integer
    path,          'string; only used if extlink is TRUE
    macro_to_run, 'string: name of elf macro
    no_border     'Boolean
```

SS_SET_OBJECT_HIDDEN@

Sets if an object is hidden

Format SS_SET_OBJECT_HIDDEN@(objName, hideFlag)

Method [this.set_object_hidden@](#)(objName, hideFlag)

Arguments

objName	The name of the object whose <i>hidden</i> state is being set.
---------	--

hideFlag A Boolean value which if set to TRUE indicates that the object will be hidden (that is, made invisible).

Description If hideFlag is set to TRUE, sets a state flag within the object that indicates that the object will be made invisible. If this value is FALSE, the object is made visible.

See also [SS SET OBJECT LOCKED@](#)
[SS SET OBJECT PRINT@](#)

SS_SET_OBJECT_LOCKED@

Locks or unlocks an object

Format SS_SET_OBJECT_LOCKED@(objName, lockedFlag)

Method [this.set_object_locked@\(objName, lockedFlag\)](#)

Arguments objName The name of the object whose *lock* state is being set.
lockedFlag A Boolean value which if set to TRUE indicates that the object cannot be selected.

Description If lockedFlag is set to TRUE, sets a state flag within the object that indicates that the object is locked; that is, the object cannot be selected and opened. If this value is FALSE, the object can be selected and opened.

See also [SS SET OBJECT HIDDEN@](#)
[SS SET OBJECT PRINT@](#)

SS_SET_OBJECT_PRINT@

Sets or unsets an objects print property

Format SS_SET_OBJECT_PRINT@(name, printFlag)

Method [this.set_object_print@\(name, printFlag\)](#)

Arguments name The name of the object whose *print* state is being set.
printFlag A Boolean value which if set to TRUE indicates that the object will be printed when the Spreadsheets document is printed.

Description If printFlag is set to TRUE, sets a state flag within the object that indicates that the object can be printed. If this value is FALSE, the object will not be printed when the Spreadsheets document is printed.

See also [SS_SET_OBJECT_HIDDEN@](#)
[SS_SET_OBJECT_LOCKED@](#)

SS_SET_OPEN_CELL_FOR_RNG@

Use cell attributes instead of range attributes

Format SS_SET_OPEN_CELL_FOR_RNG@(useOpenCellForRangesFlag)

Method [this.set_open_cell_for_rng@\(useOpenCellForRangesFlag\)](#)

Arguments useOpenCellForRangesFlag
A Boolean value where TRUE indicates use the open cell.

Description If attributes are being set, indicates that the open (or marked) cell's attributes are to be used instead of the attributes of the entire range. Setting this value will increase the performance of macros that set attributes.

SS_SET_OPEN_CELL_FOR_RNG_PROF@

Use cell attributes instead of range attributes

Format SS_SET_OPEN_CELL_FOR_RNG_PROF@(useOpenCellFlag)

Method [this.set_open_cell_for_rng_prof@\(useOpenCellFlag\)](#)

Arguments useOpenCellForRangesFlag
A Boolean value where TRUE indicates use the open cell.

Description If attributes are being set, indicates that the open (or marked) cell's attributes are to be used instead of the attributes of the entire range. Setting this value will increase the performance of macros that set attributes.

SS_SET_PAGEBREAKS@

Inserts a manual page break

Format SS_SET_PAGEBREAKS@(position)

Method [this.set_pagebreaks@\(position\)](#)

Arguments position A string indicating the place in a spreadsheet where a page break should be inserted. position can be a row, column, or combination of rows and columns. A page break will be inserted before the row or column you specify in position.

For example, if position is specified as "C," a page break is inserted before column C. If position is specified as "3," a page break is inserted before row 3. If position is specified as "C,3," a page break is inserted before column C and another page break is inserted before row 3.

Description Page breaks occur when you print the spreadsheet. Any number of page breaks may be specified in position. When you set a manual page break, natural page breaks are automatically adjusted to account for the manual page break.

See also [SS_CLEAR_PAGEBREAKS@](#)
[SS_SELECT_PAGEBREAKS@](#)
[SS_GET_PAGEBREAKS@](#)

SS_SET_PAGE_SETUP@

Sets print setup attributes

Format SS_SET_PAGE_SETUP@(format ss_page_setup_ newSetup)

Method [this.set_page_setup@\(format ss_page_setup_ newSetup\)](#)

Arguments newSetup A variable indicating the attributes to set for the Spreadsheets document. The ss_page_setup_ format is contained within spsheet_.am. The definition of this format is as follows:

format ss_page_setup_
width, 'The width of the page in inches
height, 'The height of the page in inches
lmargin, 'left margin
rmargin, 'right margin

tmargin, 'top margin
 bmargin, 'bottom margin
 landscape, 'TRUE if landscape
 center_halign, 'TRUE if center aligned horizontally
 center_valign, 'TRUE if center aligned vertically
 prt_headers, 'TRUE if print row and column headings
 facing_pages, 'TRUE if facing pages are printed
 print_to_fit, 'TRUE if print to fit
 'If print_to_fit, the number of pages
 'wide and tall
 num_pages_wide,
 num_pages_tall,
 'TRUE if borders and shading beyond the last cell con-
 taining data are not printed
 no_print_beyond_last_cell,
 print_gridlines 'TRUE if to print grid lines
 paper_type A value from 0 to 17, as follows:
 0 US Letter
 1 Tabloid
 2 Ledger
 3 Legal
 4 Statement
 5 Executive
 6 Envelope 10
 7 Envelope 9
 8 Envelope 6
 9 A3
 10 A4
 11 A5
 12 B4
 13 B5
 14 Envelope C4
 15 Envelope C5
 16 Envelope DIN
 17 Custom

Description Sets the print setup specified for the current spreadsheet.

See also [SS_GET_PAGE_SETUP@](#)

SS_SET_PROTECTION@

Protects spreadsheet cells and range from editing

Format SS_SET_PROTECTION@(cellOrRange)

Method [this.set_protection@](#)(cellOrRange)

Arguments cellOrRange A string indicating the cell address or the range of cells being protected.

Description Prevents spreadsheet cells and ranges from being edited. Protected cells cannot be deleted. They can, however, be filled, using [SS_FILL@](#) or by using [SS_PUT_CELL@](#) with its force argument set to TRUE.

See also [SS_CLEAR_PROTECTION@](#).

SS_SET_PT_SIZE@

Sets the point size

Format SS_SET_PT_SIZE@(size)

Method [this.set_pt_size@](#)(size)

Arguments size One of the following values: 6, 8 10, 12, 14, 18, 24, 36.

Description Sets the point size of the cells in the current selection to size.

SS_SET_RANGE_ATTR@

Sets a range's text attributes

Format SS_SET_RANGE_ATTR@(ranges, color, boldFlag, italicFlag, underline, face, ptSize, resetToDefaultFlag)

Method [this.set_range_attr@](#)(ranges, color, boldFlag, italicFlag, underline, face, ptSize, resetToDefaultFlag)

Arguments ranges The ranges whose attributes are being set.

color The string name of the range's color.

boldFlag	A Boolean value which if set to TRUE indicates that information will be displayed in bold.
italicFlag	A Boolean value which if set to TRUE indicates that information will be displayed in italics.
underline	A numeric value indicating if information is underlined, and, if it is, how many underlines, as follows: 0 No underline 1 Single underline 2 Double underline
face	A string indicating the font used to display information.
ptSize	A number indicating the point size to be used when displaying information. Values you can use are 6, 8, 10, 12, 14, 18, 24, and 36.
resetToDefaultFlag	A Boolean value which if set to TRUE resets all of these attributes to their default values.

Description Sets the text attributes for a range to the indicated values. If `resetToDefaultFlag` is TRUE, these attributes are set to their default values. In this case, any values you may pass as parameters to other arguments are ignored.

SS_SET_RANGE_PT_SIZE@

Sets a range's point size

Format `SS_SET_RANGE_PT_SIZE@(range, size)`

Method `this.set_range_pt_size@(range, size)`

Arguments

range	The range whose point size is being set.
size	The point size, which can be one of the following values: 6, 8, 10, 12, 14, 18, 24, or 36.

Description The point size at which information is displayed to size within the named range.

See also [**SS_GET_FONT_SIZES@**](#)
[**SS_SET_CELL_FONT_SIZE@**](#)

SS_SET_RECORD_MODE@

Sets the recording mode of the Record Macro function

Format SS_SET_RECORD_MODE@(mode)

Method [this.set_record_mode@\(mode\)](#)

Arguments mode The recording mode to use when recording keystrokes in a Spreadsheet:

- 0 **Absolute recording:** records literal cell addresses and no carriage returns. During playback, keystrokes occur in exactly the same cells as when recorded.
- 1 **Relative recording:** records relative cell addresses. During playback, keystrokes occur relative to the current cursor position.

Description Allows you to toggle between absolute and relative recording modes for the Spreadsheets keystroke recorder.

This macro affects only the current Applixware session; it does not affect your ax_prof4 file. To set the recording mode in your ax_prof4 file, use SS_SET_RECORD_MODE_PROFILE@, which is called by the * ® Spreadsheets Preferences menu option.

See also [SS_RECORD_MACRO@](#)
[SS_SET_RECORD_MODE_PROFILE@](#)

SS_SET_RECORD_MODE_PROFILE@

Indicates if absolute or relative addresses are recorded

Format SS_SET_RECORD_MODE_PROFILE@ (recordModeFlag)

Method [this.set_record_mode_profile@ \(recordModeFlag\)](#)

Arguments recordModeFlag

A Boolean value which if set to TRUE indicates that cell addresses will be recorded as relative rather than absolute when recording macros.

See also [SS_RECORD_MACRO@](#)
[SS_SET_RECORD_MODE@](#)

SS_SET_ROWCOL_HEADINGS@

If true, displays row and column headings

Format SS_SET_ROWCOL_HEADINGS@(flag)

Method [this.set_rowcol_headings@\(flag\)](#)

Arguments flag A Boolean value which if set to TRUE indicates that row and column headings will be displayed.

SS_SET_ROW_HEIGHT@

Sets the height for one or more rows

Format SS_SET_ROW_HEIGHT@(rows, height, useStdHeightFlag)

Method [this.set_row_height@\(rows, height, useStdHeightFlag\)](#)

Arguments rows The rows whose height is being is set.
height The height to which the rows are being set. The minimum row height is 6 points. The maximum row height is 200 points.
useStdHeightFlag A Boolean value which if set to TRUE indicates that rows will be set to their default value. In this case, height is ignored.

See also [SS_GET_ROW_HEIGHT@](#)
[SS_SET_WIDTH@](#)

SS_SET_ROWS_RESIZE_PROFILE@

Adjusts cell height as text wraps

Format SS_SET_ROWS_RESIZE_PROFILE@ (autoResizeRowsFlag)

Method [this.set_rows_resize_profile@ \(autoResizeRowsFlag\)](#)

Arguments autoResizeRowsFlag A Boolean value which if set to TRUE indicates that Applixware

Spreadsheets will automatically adjust the row size to the height of cells containing wrapped text.

SS_SET_SCREEN_TOP_LEFT@

Sets the top left-hand cell of the screen

Format SS_SET_SCREEN_TOP_LEFT@(cell)

Method [this.set_screen_top_left@](#)(cell)

Method [this.set_screen_top_left@](#) (cell)

Arguments cell An ElfData value indicating which cell to display in the upper lefthand corner of the screen.

Description Sets the cell indicated in the argument in the upper lefthand corner of the screen.

SS_SET_SHEET@

Makes a sheet active

Format SS_SET_SHEET@(sheetName)

Method [this.set_sheet@](#)(sheetName)

Arguments sheetName The name of a sheet in the current spreadsheet.

Description Makes sheetName the active sheet.

SS_SET_SHEETS_DISPLAY@

Sets the display of multiple sheets

Format SS_SET_SHEETS_DISPLAY@(flag)

Method [this.set_sheets_display@](#)(flag)

Arguments flag A Boolean value which if set to TRUE turns on the display of multiple sheets. FALSE sets the display of a single sheet.

Description Sets the display of multiple sheets.

SS_SET_STRUCTURED_COMMENT@

Sets a structured comment in an .as file

Format SS_SET_STRUCTURED_COMMENT@
(commentName, commentString)

Method [this.set_structured_comment@](#)
(commentName, commentString)

Arguments commentName
A string containing a structured comment within a Spreadsheets document.

Description Adds a text comment into Spreadsheets .as file. This comment is invisible from Applixware Spreadsheets, but you can see the structured comment if you use the more command on the file from UNIX. The following shows an example of a macro that embeds a structured comment:

```
macro ss_set_struc_com
var com
com=ss_set_structured_comment@("CommentName", com)
endmacro
```

The following shows two lined containing a structured comment as it would appears in a .as file.

```
*BEGIN SPREADSHEETS VERSION=420/420 ENCODING=7BIT
** "CommentName" CommentValue
```

See also [SS_GET_STRUCTURED_COMMENT@](#)

SS_SET_TITLES@

Makes a row or column into a title row or column

Format SS_SET_TITLES@(list)

Method [this.set_titles@](#)(list)

Arguments list
A string indicating the rows and columns to be made into title rows and columns.

Description When a column is made into a title, it is displayed on the left side of the screen and does not scroll to the left or right. When a row is made into a title, it is displayed at the top of the screen and does not scroll up or down. Cells in title rows and columns cannot be edited.

If multiple rows or columns are made into titles, they are displayed in numeric or alphabetic order, regardless of the order in which they were created.

For example, if you use `SS_SET_TITLES@` to make rows 5 and 6 into titles, and then use the macro again to make row 3 into a title, the titles are displayed at the top of the screen in the order of 3 followed by 5 followed by 6.

Description If a specified row or column in a list is already a title, it remains a title. Specify a null string to clear all titles.

SS_SET_VIEW_FORMULAS@

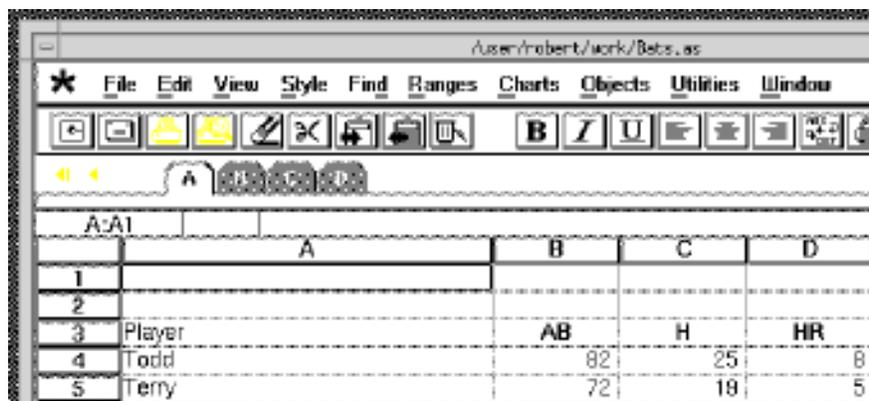
Displays formulas in Spreadsheet cells

Format `SS_SET_VIEW_FORMULAS@(viewFlag)`

Method `this.set_view_formulas@(viewFlag)`

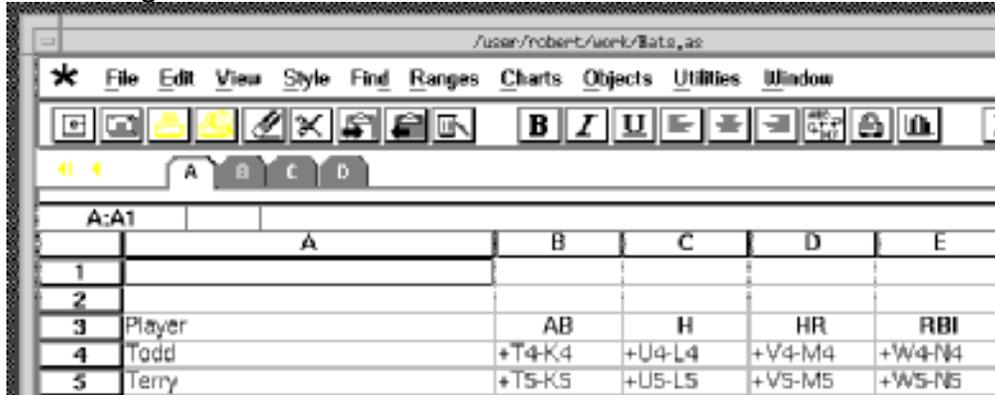
Arguments `viewFlag` A boolean. If set to TRUE, formulas are displayed in the cells of the current spreadsheet. If set to FALSE, values are displayed.

Description Turns the display of formulas in the Spreadsheets on and off. If `viewFlag` is set to FALSE, the spreadsheet displays values in the cells, as shown in the following illustration:



	A	B	C	D
1				
2				
3	Player	AB	H	HR
4	Todd	82	25	8
5	Terry	72	18	5

If viewFlag is set to TRUE, the spreadsheet displays formulas in the cells, as shown in the following illustration:



SS_SET_WIDTH@

Sets column widths

Format SS_SET_WIDTH@(col, width)

Method [this.set_width@\(col, width\)](#)

Arguments

col	A string indicating the columns for which the width should be set. The column letters should be separated by commas. For example, to set the widths for columns A, B, and C, you supply the columns argument "A,B,C". A range of columns can be specified as beginning column-ending column. For example, columns A through F can be specified as A-F.
width	A number indicating how many characters wide to make the columns specified by columns.

Description Sets the column width for the indicated columns. The default width is 10 characters. The minimum allowable width is one character and the maximum allowable width is 60 characters.

See also [SS_GET_COL_WIDTH@](#)
[SS_SET_ROW_HEIGHT@](#)

SS_SET_WORKSHEET_NAME@

Sets the name of a worksheet

Format SS_SET_WORKSHEET_NAME@(sheetNumber, name)

Method [this.set_worksheet_name@\(sheetNumber, name\)](#)

Arguments sheetNumber The number of the worksheet whose name you want to set. This number is zero-based.
name A string that becomes the new name of the worksheet.

Description Sets the name of a Spreadsheets worksheet. After you change the name of a worksheet, the new name should appear when you select Edit ® Rename Sheets from the Spreadsheets menu.

The sheetNumber argument is zero-based. In a default worksheet, the sheet names and numbers are as follows:

0 is A

1 is B

2 is C

3 is D

See also [SS_GET_WORKSHEET_NAME@](#)

SS_SET_ZEROVAL_PROFILE@

Sets your profile to indicate if items that have a zero value are displayed

Format SS_SET_ZEROVAL_PROFILE@(zeroFlag)

Method [this.set_zeroval_profile@\(zeroFlag\)](#)

Arguments zeroFlag A Boolean value which if set to TRUE tells Spreadsheets to display zero values when they occur as the last digit to the right of the decimal point or when the value of a cell is 0.
If this option FALSE, zero values are not displayed when they occur as the last digit to the right of the decimal point or when the value of a cell is 0.

See also [SS_SET_ZERO_VALS@](#)

SS_SET_ZERO_VALS@

Sets how zero values are displayed

Format SS_SET_ZERO_VALS@(flag)

Method [this.set_zero_vals@\(flag\)](#)

Arguments flag A Boolean value which if set to TRUE tells Spreadsheets to display zero values when they occur as the last digit to the right of the decimal point or when the value of a cell is 0.
If this option FALSE, zero values are not displayed when they occur as the last digit to the right of the decimal point or when the value of a cell is 0.

See also [SS_SET_ZEROVAL_PROFILE@](#)

SS_SET_ZOOM_FACTOR@

Sets the zoom of the Spreadsheets window in percent

Format SS_SET_ZOOM_FACTOR@(percent)

Method [this.set_zoom_factor@\(percent\)](#)

Arguments percent A value representing the zoom factor in the Spreadsheets window.

SS_SHEET_NUM@

Converts a worksheet letter string to a worksheet number

Format sheetNum = SS_SHEET_NUM@(sheetString)

Method [this.sheet_num@\(sheetString\)](#)

Arguments sheetString The letter representation of a worksheet, followed by a comma.

Description Returns the number that corresponds to the supplied letter representation of a worksheet. The returned number is zero-based. For example, in an empty worksheet, the statement `SS_SHEET_NUM@("A:")` returns a zero, the statement `SS_SHEET_NUM@("B:")` returns a 1, and so on. This macro does not work with user-defined worksheet names.

See also [SS_SHEET_STRING@](#)

SS_SHEET_STRING@

Converts a worksheet number to a worksheet letter string

Format sheetNum = SS_SHEET_STRING@(sheetNumber)

Method [this.sheet_string@](#)(sheetNumber)

Arguments sheetNumber The zero-based numerical representation of a worksheet

Description Returns the string name that corresponds to the supplied worksheet number. The supplied number is zero-based. For example, in an empty worksheet, the statement SS_SHEET_STRING@(0) returns the string "A:", the statement SS_SHEET_STRING@(1) returns "B:", and so on. This macro does not return user-defined worksheet names.

See also [SS_SHEET_NUM@](#)

SS_SHOW_PAGEBREAKS@

Toggles between hiding and showing page breaks

Format SS_SHOW_PAGEBREAKS@()

Method [this.show_pagebreaks@](#)

Description Toggles the display of page breaks. If page breaks are being displayed, the appropriate page number displays in the status line when you click on any cell. The page break will appear as a dashed line.

Page breaks display according to the visible rows and columns in your spreadsheet. For example, if there is a vertical page break after column F and you hide column C, the vertical page break will now be after column G. Manual page breaks, however, are not affected by hiding rows or columns.

This macro is bound to View ® Page Breaks.

SS_SMALLER_KEY@

Makes the width of a column 1 character narrower

Format SS_SMALLER_KEY@()

Method [this.smaller_key@](#)

SS_SORT_COLS@

Sorts the specified spreadsheet columns

Format SS_SORT_COLS@(primary, [secondary], range[, descendFlag])

Method [this.sort_cols@\(primary, \[secondary\], range\[, descendFlag\]\)](#)

Arguments

primary	A string indicating the row that will serve as the primary sort key. For example, if primary is specified as "3," and row 3 contains salary data, then the columns are sorted according to salary.
secondary	A string argument indicating the column that will serve as the secondary sort key. The secondary sort key is used for sorting items which are not unique to the primary sort key. For example, if the primary sort key row, row 3, contains salary data and the secondary sort key column, row 1, contains name data, all columns having the same salary would then be sorted according to name. If you don't want to use a secondary sort key, specify NULL as the secondary argument.
range	A string indicating the range of cells to sort.
descendFlag	Indicates whether to sort the columns in descending or ascending order. If descendFlag is set to TRUE, columns are sorted in descending (z-a, 10-1) order. If descend is set to FALSE, columns are sorted in ascending order (a-z, 1-10). The default for descendFlag is FALSE.

Description Sorts data in the specified columns according to the sort key row specifications. The use of a secondary sort key is optional. If the data includes both numbers and labels, numbers appear before labels in the sorted result.

See also [SS_SORT_RANGE@](#)
[SS_SORT_ROWS@](#)

SS_SORT_RANGE@

Sorts a range by rows or columns

Format SS_SORT_RANGE@(sortRange, sortType, firstKey[, key1DescendFlag[, secondKey[, key2DescendFlag[, thirdKey[, key3DescendFlag]]]]])

Method [this.sort_range@](#)(sortRange, sortType, firstKey[, key1DescendFlag[, secondKey[, key2DescendFlag[, thirdKey[, key3DescendFlag]]]]])

Arguments

sortRange	A string indicating the range of cells to sort.
sortType	One of the following values: 0 Sort by rows Anything else Sort by columns
firstKey	The string value of the row or column that is the primary (or first) sort key.
key1DescendFlag	A Boolean value which if set to TRUE indicates that the information indicated by firstKey is sorted in descending order.
secondKey	The string value of the row or column that is the secondary sort key.
key2DescendFlag	A Boolean value which if set to TRUE indicates that the information indicated by secondKey is sorted in descending order.
thirdKey	The string value of the row or column that is the tertiary (or third) sort key.
key3DescendFlag	A Boolean value which if set to TRUE indicates that the information indicated by thirdKey is sorted in descending order.

Description Sorts the indicated range by rows or by columns (depending on the value of sortType).

See also [SS_SORT_COLS@](#)
[SS_SORT_ROWS@](#)

SS_SORT_ROWS@

Sorts the specified spreadsheet rows

Format SS_SORT_ROWS@(primary, [secondary], range[, descendFlag])

Method `this.sort_rows@(primary, [secondary], range[, descendFlag])`

Arguments

<code>primary</code>	A string indicating the row that will serve as the primary sort key. For example, if <code>primary</code> is specified as "C," and row C contains zip code data, then the rows will be sorted according to zip code.
<code>secondary</code>	A string argument indicating the row that will serve as the secondary sort key. The secondary sort key is used for sorting items which are not unique to the primary sort key. For example, if the primary sort key row, row C, contains zip code data and the secondary sort key column, row A, contains name data, all rows having the same zip code would then be sorted according to name. If you don't want to use a secondary sort key, specify NULL as the secondary argument.
<code>range</code>	A string indicating the range of cells to sort.
<code>descendFlag</code>	Indicates whether to sort the rows in descending or ascending order. If <code>descend</code> is set to TRUE, rows are sorted in descending (z-a, 10-1) order. If <code>descend</code> is set to FALSE, rows are sorted in ascending order (A-Z, 1-10). The default for <code>descend</code> is FALSE.

Description Sorts data in the specified rows according to the sort key row specifications. The use of a secondary sort key is optional. If the data includes both numbers and labels, numbers appear before labels in the sorted result.

See also [SS_SORT_COLS@](#)
[SS_SORT_RANGE@](#)

SS_SPECIAL_ENTER_KEY@

Enters array data

Format `SS_SPECIAL_ENTER_KEY@()`

Method `this.special_enter_key@`

Description Inserts an array of information into a range. Normally, this macro is bound to CTRL-SHIFT-RETURN. If this is not convenient, use this macro to bind the array insertion to another key chord.

SS_SUM@

Inserts the SS_SUM function on the edit line

Format SS_SUM@()

Description Inserts the following string on the Spreadsheets edit line:

SS_SUM@()

The cursor is set just after the first parenthesis. You can either type in a range of cells to add, or select a range of cells with your mouse.

SS_TAB_KEY@

Moves one screen to the right

Format SS_TAB_KEY@()

Method [this.tab_key@](#)

See also [SS_RIGHT_SCREEN_KEY@](#)

SS_TEXT_?@

Applies the specified typeface to selected text

Format SS_TEXT_?@()

Method [this.text_?@](#)

Description SS_TEXT_?@ is called by the Style ® Character Settings option. The specified typeface becomes the default. The following typeface macros are supported:

SS_TEXT_AVANT_GARDE@

SS_TEXT_BOOKMAN@

SS_TEXT_CHANCERY@

SS_TEXT_COURIER@

SS_TEXT_DEFAULT@

SS_TEXT_DINGBATS@

SS_TEXT_GOTHIC@
SS_TEXT_HELV_NARROW@
SS_TEXT_MINCHOU@
SS_TEXT_MONOSPACE@
SS_TEXT_HELVETICA@
SS_TEXT_PALATINO@
SS_TEXT_SANS_SERIF@
SS_TEXT_SCHOOLBOOK@
SS_TEXT_SERIF@
SS_TEXT_SYMBOL@
SS_TEXT_TIMES@

SS_TEXT_?PT@

Sets the text's point size

Format SS_TEXT_?PT@()

Method [this.text_?pt@](#)

Description Sets the display and print size for the current selection to the indicated point size.

SS_TEXT_10PT@
SS_TEXT_12PT@
SS_TEXT_14PT@
SS_TEXT_18PT@
SS_TEXT_24PT@
SS_TEXT_36PT@
SS_TEXT_6PT@
SS_TEXT_8PT@

SS_TOGGLE_SYNC_SCROLL@

Toggles the synchronous scrolling between multiple windows

Format SS_TOGGLE_SYNC_SCROLL@()

Method [this.toggle_sync_scroll@](#)

SS_TOP_SECTION@

Moves the cursor up to the next cell containing data

Format SS_TOP_SECTION@()

Method [this.top_section@](#)

Description Moves the cursor from its current position to the next cell up that contains data. If no cells containing data are found above the current cursor position, the cursor is moved to the first cell in the column. SS_TOP_SECTION@ is called by the Keys ® Next data up menu option.

See also [SS_BOTTOM_SECTION@](#)

SS_TYPE@

Inserts a string at the edit line's edit cursor

Format SS_TYPE@(string)

Method [this.type@\(string\)](#)

Arguments string The text to be inserted in the edit line.

SS_UNDERLINE@

Toggles the underline attribute to be on or off

Format SS_UNDERLINE@()

Method [this.underline@](#)

See also [SS_BOLD@](#)
[SS_ITALICS@](#)

SS_UNDO@

Undoes the last action in the current Spreadsheets document

Format SS_UNDO@()

Method [this.undo@](#)

Description Not all actions can be undone. For example, you cannot undo a save. SS_UNDO@ is called by the Edit ® Undo menu option.

SS_UNINSTALL_ADDIN_FUNCTIONS@

Detaches C library; auto rebind can still occur

Format SS_UNINSTALL_ADDIN_FUNCTIONS@()

Method [this.uninstall_addin_functions@](#)

Description Detaches the libss.so.1.1 library from the current process. When the library was bound to your process, information was recorded for the functions within this library. While these references are removed, it is still possible to automatically bind to these add-in functions.

SS_UNREGISTER_FUNCTION@

Deletes a function from the Spreadsheets Functions box

Format SS_UNREGISTER_FUNCTION@(registerName)

Method [this.unregister_functions@\(registerName\)](#)

Arguments registerName The name under which the function was registered with Applixware. You register functions with Applixware using the macro **SS_REGISTER_FUNCTION@**.

Description Deletes a user-defined built-in function from the Spreadsheets Functions list. This function must have been registered with the SS_REGISTER_FUNCTION@ macro.

The registerName argument must be identical to the second argument of the SS_REGISTER_FUNCTION@ macro that added the function to the Spreadsheets

function box. For example, suppose you register a function from the RUN MACRO box with this command:

```
SS_REGISTER_FUNCTION@ foo bar foo()
```

To unregister the function, use this command:

```
UNREGISTER_FUNCTION@ bar
```

See also [SS REGISTER FUNCTION@](#)

SS_UNLOCK_SHEET@

Unlocks a sheet in a Spreadsheets document

Format SS_UNLOCK_SHEET@(sheet, passwd)

Arguments

sheet	The sheet letter. This sheet will be locked.
passwd	A string. This password must match the password used to lock the sheet. The password is case sensitive. It is stored in plain text in the Spreadsheets document when the file is saved.

See also [SS LOCK SHEET@](#)

SS_UNSET_CALLBACK@

Disables a Spreadsheet callback function

Format SS_UNSET_CALLBACK@(macro)

Method [this.unset_callback@\(macro\)](#)

Arguments

macro	The macro to be disabled that previously was enabled via SS_SET_CALLBACK@.
-------	--

Description SS_UNSET_CALLBACK@ turns off a Spreadsheet callback function that was invoked using the SS_SET_CALLBACK@ macro. SS_UNSET_CALLBACK@ can be called at any time, even from within the callback macro. For example, you could design a callback macro such that it would first trigger one macro, then disable that callback, and then enable a different callback.

See also [SS SET CALLBACK@](#)
[SS SET CR CALLBACK@](#)
[SS UNSET CR CALLBACK@](#)

[SS SET DBL CLICK CALLBACK@](#)

[SS UNSET DBL CLICK CALLBACK@](#)

[SS SET LOAD CELL CALLBACK@](#)

[SS UNSET LOAD CELL CALLBACK@](#)

SS_UNSET_CR_CALLBACK@

Disables a Spreadsheet callback function

Format SS_UNSET_CR_CALLBACK@(macro)

Method [this.unset_cr_callback@\(macro\)](#)

Arguments macro The macro to be disabled that previously was enabled using SS_SET_CR_CALLBACK@.

Description Turns off a Spreadsheet callback function that was invoked using the SS_SET_CR_CALLBACK@ macro. SS_UNSET_CR_CALLBACK@ can be called at any time, even from within the callback macro. For example, you could design a callback macro such that it would first trigger one macro, then disable that callback, and then enable a different callback.

See also [SS SET CR CALLBACK@](#)
[SS SET CALLBACK@](#)
[SS UNSET CALLBACK@](#)
[SS SET DBL CLICK CALLBACK@](#)
[SS UNSET DBL CLICK CALLBACK@](#)
[SS SET LOAD CELL CALLBACK@](#)
[SS UNSET LOAD CELL CALLBACK@](#)

SS_UNSET_DBL_CLICK_CALLBACK@

Disables a Spreadsheet callback function

Format SS_UNSET_DBL_CLICK_CALLBACK@(macro)

Method [this.unset_dbl_click_callback@\(macro\)](#)

Arguments macro The macro to be disabled that previously was enabled using SS_SET_DBL_CLICK_CALLBACK@.

Description Turns off a Spreadsheet callback function that was invoked using the `SS_SET_DBL_CLICK_CALLBACK@` macro. `SS_UNSET_DBL_CLICK_CALLBACK@` can be called at any time, even from within the callback macro. For example, you could design a callback macro such that it would first trigger one macro, then disable that callback, and then enable a different callback.

See also [SS SET DBL CLICK CALLBACK@](#)
[SS SET CALLBACK@](#)
[SS UNSET CALLBACK@](#)
[SS SET CR CALLBACK@](#)
[SS UNSET CR CALLBACK@](#)
[SS SET LOAD CELL CALLBACK@](#)
[SS UNSET LOAD CELL CALLBACK@](#)

SS_UNSET_LOAD_CELL_CALLBACK@

Disables a Spreadsheet selection callback function

Format `SS_UNSET_LOAD_CELL_CALLBACK@(macro)`

Method `this.unset_load_cell_callback@(macro)`

Arguments macro The macro to be disabled that previously was enabled using `SS_SET_LOAD_CELL_CALLBACK@`.

Description Disables a Spreadsheet callback function that was invoked using the [SS SET LOAD CELL CALLBACK@](#) macro.

See also [SS SET CALLBACK@](#)
[SS UNSET CALLBACK@](#)
[SS SET CR CALLBACK@](#)
[SS UNSET CR CALLBACK@](#)
[SS SET DBL CLICK CALLBACK@](#)
[SS UNSET DBL CLICK CALLBACK@](#)

SS_UPDATE_REFS@

Updates all links needing updating

Format `SS_UPDATE_REFS@()`

Method `this.update_refs@`

Description Updates any links that need to be updated. This macro is bound to Ranges ® Update External Links.

SS_UP_ARROW_KEY@

Moves the cell pointer up from the current cell

Format SS_UP_ARROW_KEY@()

Method [this.up_arrow_key@](#)

See also [SS BACK RETURN KEY@](#)
[SS DOWN ARROW KEY@](#)
[SS LEFT ARROW KEY@](#)
[SS RETURN KEY@](#)
[SS RIGHT ARROW KEY@](#)

SS_UPDATE_CHARTS@

Updates all charts in the current spreadsheet

Format SS_UPDATE_CHARTS@(chartName)

Method [this.update_charts@\(chartName\)](#)

Arguments chartName A string name of a chart in the spreadsheet

Description Updates all the charts in the current spreadsheet to reflect the current data. This macro runs when you select Charts ® Update from the Spreadsheets menu.

SS_VIEW_EXPRESSLINE@

Toggles the display of the ExpressLine

Format SS_VIEW_EXPRESSLINE@()

Method [this.view_expressline@](#)

Description Turns the *Expressline* on and off in Applixware Spreadsheets.

SS_VIEW_GET_COL_WIDTH@

Returns the width of a column
in a named view

Format SS_VIEW_GET_COL_WIDTH@(viewName, colString)

Method [this.view_get_col_width@\(viewName, colString\)](#)

Arguments viewName A string containing the name of a view
colString A string indicating a column in the named view

Description Returns the width of a column in a named view. You can see the named views in your spreadsheet by selecting View Y Display named view from the Spreadsheets menu bar.

SS_VIEW_GET_ROW_HEIGHT@

Returns the height of a row
in a named view.

Format SS_VIEW_GET_ROW_HEIGHT@(viewName, row_string, stdHeight)

Method [this.view_get_row_height@\(viewName, row_string, stdHeight\)](#)

Arguments viewName A string containing the name of a view
row_string A string containing the range of rows being examined
stdHeight A flag that determined the information returned from the macro.

Description If stdHeight is equal to zero, and all the rows in the specified range are the same height, SS_VIEW_GET_ROW_HEIGHT@ returns the height of the specified rows.

If stdHeight is equal to zero, and the rows in the specified range are of different heights, SS_VIEW_GET_ROW_HEIGHT@ returns -1.

If stdHeight is equal to a non-zero value, SS_VIEW_GET_ROW_HEIGHT@ returns the standard row height for the named view.

SS_VIEW_HIDE@

Hides a portion of a named view

Format SS_VIEW_HIDE@(viewName, rangeString)

Method [this.view_hide@](#)(viewName, rangeString)

Arguments viewName A string containing the name of a view in the current spreadsheet.
rangeString A string containing a set of columns or rows to be hidden.

Description Hides rows or columns in a named view in the spreadsheet. The viewName parameter contains the name of a named view. The rangeString contains one of the following:

- A row designation. For example, SS_VIEW_HIDE@("MyView:", "1") hides row 1.
- A column designation. For example, SS_VIEW_HIDE@("MyView:", "A") hides column A.
- A set of contiguous rows. For example, SS_VIEW_HIDE@("MyView:", "1..3") hides rows 1 through 3.
- A set of contiguous columns. For example, SS_VIEW_HIDE@("MyView:", "A..C") hides columns A through C.

See also [SS_VIEW_UNHIDE@](#)

SS_VIEW_LOAD_RANGE@

Displays only the selected cells

Format SS_VIEW_LOAD_RANGE@()

Method [this.view_load_range@](#)

Description Creates a spreadsheet view of all currently selected cells. Cells that aren't selected are hidden from view. If no cells are selected, only the cell in which the cursor resides will be displayed. If multiple ranges are selected, only the first contiguous range encountered is displayed in the view.

See also [SS_HIDE@](#)
[SS_REVEAL@](#)
[SS_REVEAL_SOME@](#)

[SS_VIEW_RANGE@](#)

SS_VIEW_NAMES@

Returns an array containing the names of all named views, including sheet names

Format viewNameArray = SS_VIEW_NAMES@()

Method viewNameArray = this.view_names@

Description Returns an array in which each element of the array is the name of a named view in the current spreadsheet. If no named views have been defined for the spreadsheet, NULL is returned.

SS_VIEW_RANGE@

Displays the specified range

Format SS_VIEW_RANGE@(range)

Method this.view_range@(range)

Arguments range Indicates the range to display. range can be a range string specification or a single cell specification. Note, however, that range cannot be a range name.

Description Displays the specified range in the current spreadsheet. All cells not in range are hidden.

See also [SS_HIDE@](#)
[SS_REVEAL@](#)
[SS_REVEAL_SOME@](#)
[SS_VIEW_LOAD_RANGE@](#)

SS_VIEW_SELECTED@

Creates a view that includes only the specified rows and/or columns

Format SS_VIEW_SELECTED@(rowOrCol)

Method `this.view_selected@(rowOrCol)`

Arguments `rowOrCol` A string indicating the rows or columns to display. `rowOrCol` can be a single row or column, or a range of rows or columns specified in the form beginning range-end range. For example, to create a view containing columns B through F, specify B-F.

Description Displays only the rows or columns specified in the current spreadsheet. All other rows and columns are hidden from view.

SS_VIEW_UNHIDE@

Reveals a portion of a named view

Format `SS_VIEW_UNHIDE@(viewName, rangeString)`

Method `this.view_unhide@(viewName, rangeString)`

Arguments `viewName` A string containing the name of a view in the current spreadsheet.
`rangeString` A string containing a set of columns or rows to be hidden.

Description Reveals rows or columns in a named view in the spreadsheet. The `viewname` parameter contains the name of a named view. The `rangeString` contains one of the following:

- A row designation. For example, `SS_VIEW_UNHIDE@("MyView:", "1")` reveals row 1.
- A column designation. For example, `SS_VIEW_UNHIDE@("MyView:", "A")` reveals column A.
- A set of contiguous rows. For example, `SS_VIEW_UNHIDE@("MyView:", "1..3")` reveals rows 1 through 3.
- A set of contiguous columns. For example, `SS_VIEW_UNHIDE@("MyView:", "A..C")` reveals columns A through C.

See also [SS_VIEW_HIDE@](#)

SS_VISIBILITY@

Toggles the visibility state of cells in the selection

Format `SS_VISIBILITY@()`

Method `this.visibility@`

See also [SS_VISIBLE@](#)

SS_VISIBLE_SELECTED@

SS_VISIBLE@

Makes the contents of the specified cells visible

Format SS_VISIBLE@(range)

Method [this.visible@](#)(range)

Arguments range A string indicating the range of cells that should be made visible. Multiple ranges can be specified.

Description Makes the specified cells in the current spreadsheet visible.

See also [SS_INVISIBLE@](#).
[SS_VISIBILITY@](#)

SS_VISIBLE_SELECTED@

Makes the selection visible

Format SS_VISIBLE_SELECTED@()

Method [this.visible_selected@](#)

Description Makes any cells that are invisible in the current selection visible. This macro is the same as [SS_VISIBLE@](#) except that the current range is the implicit range.

See also [SS_INVISIBLE@](#).
[SS_VISIBILITY@](#)

SS_WRAP_TEXT@

Toggles the wrap text property of cells in the selection

Format SS_WRAP_TEXT@()

Method [this.wrap_text@](#)

Description Toggles (turns on and off) the wrapping property for a cell. When the wrapping property is set, text wraps within the cell. When it is not set, text extends out to the right of the cell's boundary.

SS_ZOOM_100@

Sets the zoom factor to 100%

Format SS_ZOOM_100@()

Method [this.zoom_100@](#)

Description Sets the zoom factor in the Spreadsheets document to 100%.

SS_ZOOM_120@

Sets the zoom factor to 120%

Format SS_ZOOM_120@()

Method [this.zoom_120@](#)

Description Sets the zoom factor in the Spreadsheets document to 120%.

SS_ZOOM_150@

Sets the zoom factor to 150%

Format SS_ZOOM_150@()

Method [this.zoom_150@](#)

Description Sets the zoom factor in the Spreadsheets document to 150%.

SS_ZOOM_200@

Sets the zoom factor to 200%

Format SS_ZOOM_200@()

[Method](#) this.zoom_200@

Description Sets the zoom factor in the Spreadsheets document to 200%.

SS_ZOOM_300@

Sets the zoom factor to 300%

Format SS_ZOOM_300@()

[Method](#) this.zoom_300@

Description Sets the zoom factor in the Spreadsheets document to 300%.

SS_ZOOM_400@

Sets the zoom factor to 400%

Format SS_ZOOM_400@()

[Method](#) this.zoom_400@

Description Sets the zoom factor in the Spreadsheets document to 400%.

SS_ZOOM_40@

Sets the zoom factor to 40%

Format SS_ZOOM_40@()

[Method](#) this.zoom_40@

Description Sets the zoom factor in the Spreadsheets document to 40%.

SS_ZOOM_60@

Sets the zoom factor to 60%

Format SS_ZOOM_60@()

Method `this.zoom_60@`

Description Sets the zoom factor in the Spreadsheets document to 60%.

SS_ZOOM_80@

Sets the zoom factor to 80%

Format `SS_ZOOM_80@()`

Method `this.zoom_80@`

Description Sets the zoom factor in the Spreadsheets document to 80%.

CREATE_MAIL_SS_COPY@

Writes temporary copy of file into mail temporary directory

Format `mailNameArray = CREATE_MAIL_SS_COPY@()`

Description Writes a copy of the current Spreadsheets document into Applixware Mail's temporary directory.

This macro is one of the macros executed by Spreadsheets when you choose File ® Send. This temporary copy contains all of your current changes (even if the changes have not yet been saved to disk). Ordinarily, this temporary file is deleted after the file is mailed.

The following two-element array is returned:

`array[0,0]` The name of the temporary file.

`array[0,1]` The file's original file name.

FILTER_ASCII_TO_SS@

Converts an ASCII file into an Applixware Spreadsheets file

Format `FILTER_ASCII_TO_SS@(infile, outfile)`

Arguments infile The full path name of the ASCII file to be converted.
 outfile The full path name of the resulting Applixware Spreadsheets file, including the .as extension.

Description Creates an Applixware Spreadsheets file from an ASCII file.
For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_CSV_TO_SS@

Converts an ASCII file into an Applixware Spreadsheets file

Format FILTER_CSV_TO_SS@(infile, outfile)

Arguments infile The full pathname of the ASCII file to be converted.
 outfile The full pathname of the resulting Applixware Spreadsheets file, including the .as extension.

Description Creates an Applixware Spreadsheets file from an ASCII file contains data in *Comma Separated Values* form.
For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_DIF_TO_SS@

Converts a "Data Interchange Format" (DIF) file to an Applixware Spreadsheets file

Format FILTER_DIF_TO_SS@(infile, outfile)

Arguments infile The full pathname of the DIF file to be converted.
 outfile The full pathname of the resulting Applixware Spreadsheets file, including the .as extension.

Description Creates an Applixware Spreadsheets file from a DIF file.

Other DIF filters include:

· [FILTER_SS_TO_DIF@](#)

· [SS_IMPORT_DIF@](#)

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_SS_TO_ASCII@

Creates an ASCII file consisting of a grid of multiple columns and rows

Format FILTER_SS_TO_ASCII@(infile, outfile)

Arguments

infile	The full pathname of the Applixware Spreadsheets file from which the ASCII file will be created. infile must include the .as extension.
outfile	The full pathname of the resulting ASCII file.

Description Converts an Applixware Spreadsheets file into ASCII format. The resulting ASCII file is in the standard "grid" format, consisting of multiple rows and columns separated by the delimiter specified in the [ASCII Delimiter](#) setting in *® [Spreadsheet Preferences](#) or ([SS SET DELIMITER PROFILE@](#)). Similarly, the [Quotes Around Labels](#) preference directs this macro to place quotation arks around exported labels.

Numeric cell values can be formatted or unformatted, depending on the current [Formatted](#) setting selected in the Spreadsheet Preferences (see [SS SET ASCII FORMATTED@](#)). To change the ASCII delimiter in the *® Spreadsheet Preferences ASCII Export setting, type the ASCII value for the character you want to use. Typical ASCII delimiters include:

- 9 (horizontal tab)
- 10 (line feed)
- 35 (pound sign)
- 44 (comma).

The following attributes are lost during an ASCII export:

- Styling (bold, italic, underline)
- Typefaces (the default is Courier)
- Point sizes (the default is printer default)

Other ASCII filters include:

- [FILTER SS TO ASCII ROW@](#)
- [SS IMPORT ASCII@](#)
- [FILTER ASCII TO SS@](#)
- [FILTER SS TO ASCII COL@](#)

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_SS_TO_ASCII_COL@

Creates an ASCII file consisting of one column of multiple rows

Format FILTER_SS_TO_ASCII_COL@(infile, outfile)

Arguments infile The full pathname of the Applixware Spreadsheets file from which the ASCII file will be created. infile must include the .as extension.
outfile The full pathname of the resulting ASCII file.

Description Converts an Applixware Spreadsheets file into ASCII column format. Each cell in the Applixware Spreadsheets file is converted to one row in a single-column ASCII file. This ASCII file contains no delimiters. Numeric cell values can be formatted or unformatted, depending on the current **Formatted** setting in **Spreadsheet Preferences** or (**SS SET ASCII FORMATTED@**). Similarly, the **Quotes Around Labels** preference directs this macro to place quotation marks around exported labels.

The following attributes are lost during an ASCII export:

- Styling (bold, italic, underline)
- Typefaces (the default is Courier)
- Point sizes (the default is printer default)

Other ASCII filters include:

- **FILTER_SS_TO_ASCII@**
- **FILTER_SS_TO_ASCII_COL@**
- **FILTER_ASCII_TO_SS@**
- **SS_IMPORT_ASCII@**

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_SS_TO_ASCII_ROW@

Creates an ASCII file consisting of one row of multiple columns

Format FILTER_SS_TO_ASCII_ROW@(infile, outfile)

Arguments infile The full pathname of the Applixware Spreadsheets file from which the ASCII file will be created. infile must include the .as extension.
outfile The full pathname of the resulting ASCII file.

Description Converts an Applixware Spreadsheets file into ASCII row format. The resulting ASCII file consists of one row, with each cell value being separated by the delimiter specified in the **ASCII Delimiter** setting in **Spreadsheet Preferences** or (**SS SET DELIMITER PROFILE@**). Numeric cell values can be formatted or unformatted, depending on the current **Formatted** setting in that same menu option (**SS SET ASCII FORMATTED@**). Similarly, the **Quotes Around Labels** preference directs this macro to place quotation marks around exported labels.

To change the ASCII delimiter in the * Ý Spreadsheet Preferences' **Formatted** setting, type the ASCII value for the character you want to use. Typical ASCII delimiters include:

- 9 (horizontal tab)
- 10 (line feed)
- 35 (pound sign)
- 44 (comma)

The following attributes are lost during an ASCII export:

- Styling (bold, italic, underline)
- Typefaces (the default is Courier)
- Point sizes (the default is printer default)

Other ASCII filters include:

- **FILTER SS TO ASCII@** (ASCII grid)
- **FILTER SS TO ASCII COL@**
- **FILTER ASCII TO SS@**
SS IMPORT ASCII@

For more information about Spreadsheets filters, refer to the *Applixware Spreadsheets* manual.

FILTER_SS_TO_CSV@

Creates a "Comma Separated Value" (CSV) version of an Applixware Spreadsheets file

Format FILTER_SS_TO_CSV@(infile, outfile)

Arguments

infile	The full pathname of the Applixware Spreadsheets file from which the CSV file will be created. infile must include the .as extension.
outfile	The full pathname of the resulting CSV file.

Description Converts an Applixware Spreadsheets file to CSV format. The following attributes are lost during a CSV export:

- Styling (bold, italic, underline)
- Typefaces (the default is Courier)
- Point sizes (the default is printer default)

Other CSV filters include:

- [FILTER CSV TO SS@](#)
- [SS IMPORT CSV@](#)

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_SS_TO_DIF@

Creates a ``Data Interchange Format" (DIF) version of an Applixware Spreadsheets file

Format FILTER_SS_TO_DIF@(infile, outfile)

Arguments infile The full pathname of the Applixware Spreadsheets file from which the DIF file will be created. infile must include the .as extension.
outfile The full pathname of the resulting DIF file.

Description Converts an Applixware Spreadsheets file to DIF format.

Other DIF filters include:

- [FILTER DIF TO SS@](#)
- [SS IMPORT DIF@](#)

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_SS_TO_SYLK@

Creates a ``Symbolic Link File" (SYLK) version of an Applixware Spreadsheets file

Format FILTER_SS_TO_SYLK@(infile, outfile)

Arguments infile The full path name of the Applixware Spreadsheets file from which the SYLK file will be created. infile must include the .as extension.

outfile The full path name of the resulting SYLK file.

Description Converts an Applixware Spreadsheets file to SYLK format.

Other SYLK filters include:

- [FILTER SYLK TO SS@](#)
- [SS IMPORT SYLK@](#)

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_SS_TO_WK1@

Creates a Lotus 1-2-3 WK1 file from an Applixware Spreadsheets file

Format FILTER_SS_TO_WK1@ (infile, outfile)

Arguments infile The full path name of the Applixware Spreadsheets file from which the WK1 file will be created. infile must include the .as extension.

outfile The full path name of the resulting WK1 file.

Description Converts an Applixware Spreadsheets file to Lotus 1-2-3 WK1 format. Other Lotus filter macros include:

- [FILTER WK1 TO SS@](#)
- [FILTER WK3 TO SS@](#)
- [SS IMPORT WK3@](#)

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_SS_TO_WK3@

Creates a Lotus 1-2-3 WK3 file from an Applixware Spreadsheets file

Format FILTER_SS_TO_WK3@ (infile, outfile)

Arguments infile The full path name of the Applixware Spreadsheets file from which the WK3 file will be created. infile must include the .as extension.

outfile The full path name of the resulting WK3 file.

Description Converts an Applixware Spreadsheets file to WK3 format. For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_SS_TO_XLS@

Converts an Applixware Spreadsheet to MS Excel format

Format FILTER_SS_TO_XLS@(infile, outfile)

Arguments infile The full path name of the Applixware Spreadsheets file from which the XLS file will be created. infile must include the .as extension.
outfile The full path name of the resulting XLS file.

Description Converts an Applixware Spreadsheets file to an XLS, version 4.0 format. For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

See also [FILTER_SS_TO_XLS3@](#)

[FILTER_SS_TO_XLS4@](#)

[FILTER_SS_TO_XLS5@](#)

FILTER_SS_TO_XLS3@

Converts an Applixware Spreadsheet to MS Excel 3.0 format

Format FILTER_SS_TO_XLS3@(infile, outfile)

Arguments infile The full path name of the Applixware Spreadsheets file from which the XLS file will be created. infile must include the .as extension.
outfile The full path name of the resulting XLS file.

Description Converts an Applixware Spreadsheets file to an XLS, version 3.0 format. For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

See also [FILTER_SS_TO_XLS@](#)

[FILTER_SS_TO_XLS4@](#)

[FILTER_SS_TO_XLS5@](#)

FILTER_SS_TO_XLS4@

Converts an Applixware Spreadsheet to MS Excel 4.0 format

Format FILTER_SS_TO_XLS4@(infile, outfile)

Arguments infile The full path name of the Applixware Spreadsheets file from which the XLS file is to be created. infile must include the .as extension.
outfile The full path name of the resulting XLS file.

Description Converts an Applixware Spreadsheets file to XLS, version 4.0 format. For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

See also [FILTER_SS_TO_XLS@](#)
[FILTER_SS_TO_XLS3@](#)
[FILTER_SS_TO_XLS5@](#)

FILTER_SS_TO_XLS5@

Converts an Applixware Spreadsheet to MS Excel 5.0 format

Format FILTER_SS_TO_XLS5@(infile, outfile)

Arguments infile The full path name of the Applixware Spreadsheets file from which the XLS file is to be created. infile must include the .as extension.
outfile The full path name of the resulting XLS file.

Description Converts an Applixware Spreadsheets file to XLS, version 5.0 format. For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

See also [FILTER_SS_TO_XLS@](#)
[FILTER_SS_TO_XLS3@](#)
[FILTER_SS_TO_XLS4@](#)

FILTER_SS400_TO_SS311@

Converts an Applixware 4.x Spreadsheet to an Applixware 3.11 Spreadsheet

Format FILTER_SS400_TO_SS311@(rev4name, rev3name)

Arguments rev4name The full path name of the 4.x Applixware Spreadsheets document to be converted, including the .as extension.
rev3name The full path name of the resulting 3.11 Applixware Spreadsheets file, including the .as extension.

Description Converts an Applixware 4.x Spreadsheet document to an Applixware 3.11 Spreadsheet document.

See also [FILTER_SS420_TO_SS311@](#)

FILTER_SS420_TO_SS311@

Converts an Applixware 4.2 Spreadsheet to an Applixware 3.11 Spreadsheet

Format FILTER_SS420_TO_SS311@(rev4name, rev3name)

Arguments rev4name The full path name of the 4.2 Applixware Spreadsheets document to be converted, including the .as extension.
rev3name The full path name of the resulting 3.11 Applixware Spreadsheets file, including the .as extension.

Description Converts an Applixware 4.2 Spreadsheet document to an Applixware 3.11 Spreadsheet document.

See also [FILTER_SS400_TO_SS311@](#)

FILTER_SYLK_TO_SS@

Converts a ``Symbolic Link File" (SYLK) file to an Applixware Spreadsheets file

Format FILTER_SYLK_TO_SS@(infile, outfile)

Arguments infile The full path name of the SYLK file to be converted.

outfile The full path name of the resulting Applixware Spreadsheets file, including the .as extension.

Description Creates an Applixware Spreadsheets version from a SYLK file.

Other SYLK filters include:

- [FILTER SS TO SYLK@](#)
- [SS IMPORT SYLK@](#)

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_WK1_TO_SS@

Converts a Lotus 1-2-3 WK1 file to an Applixware Spreadsheets file

Format FILTER_WK1_TO_SS@(infile, outfile)

Arguments infile The full path name of the WK1 file to be converted.
outfile The full path name of the resulting Applixware Spreadsheets file, including the .as extension.

Description Creates an Applixware Spreadsheets version from a Lotus 1-2-3 WK1 file.

Other Lotus 1-2-3 filters include:

- [FILTER SS TO WK1@](#)
- [FILTER WK3 TO SS@](#)
- [SS IMPORT WK3@](#)
- [SS IMPORT WKS@](#)

When importing spreadsheet files which contain unsupported functions, the **Unsupported Functions** dialog box displays. There is also a **Import Problems** dialog box that will display if you have any other types of import problems such as unsupported formulas, range names and so on.

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_WK3_TO_SS@

Converts a Lotus 1-2-3 WK3 file to an Applixware Spreadsheets file

Format FILTER_WK3_TO_SS@(infile, outfile)

Arguments infile The full path name of the WK3 file to be converted.
outfile The full path name of the resulting Applixware Spreadsheets file, including the .as extension.

Description Creates an Applixware Spreadsheets version from a Lotus 1-2-3 WK3 file.
Beginning with release 4 of Applixware, this filter places one WK3 sheet upon one sheet of the outfile spreadsheet.

Other Lotus 1-2-3 filters include:

- [FILTER SS TO WK1@](#)
- [FILTER WK4 TO SS@](#)
- [SS IMPORT WK3@](#)
- [SS IMPORT WKS@](#)

When importing spreadsheet files which contain unsupported functions, the [Unsupported Functions](#) dialog box displays. There is also a [Import Problems](#) dialog box that will display if you have any other types of import problems such as unsupported formulas, range names and so on.

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_WK4_TO_SS@

Converts a Lotus 1-2-3 WK4 file to an Applixware Spreadsheets file

Format FILTER_WK4_TO_SS@(infile, outfile)

Arguments infile The full path name of the WK4 file to be converted.
outfile The full path name of the resulting Applixware Spreadsheets file, including the .as extension.

Description Creates an Applixware Spreadsheets version from a Lotus 1-2-3 WK4 file.

Other Lotus 1-2-3 filters include:

- [FILTER SS TO WK1@](#)
- [FILTER WK3 TO SS@](#)
- [SS IMPORT WK3@](#)
- [SS IMPORT WKS@](#)

When importing spreadsheet files which contain unsupported functions, the **Unsupported Functions** dialog box displays. There is also a **Import Problems** dialog box that will display if you have any other types of import problems such as unsupported formulas, range names and so on.

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_WKS_TO_SS@

Converts a WKS file to an Applixware Spreadsheet

Format FILTER_WKS_TO_SS@(infile, outfile)

Arguments

infile	The full path name of the WKS file to be converted.
outfile	The full path name of the resulting Applixware Spreadsheets file, including the .as extension.

Description Creates an Applixware Spreadsheets version from a Lotus 1-2-3 WKS file.

Other Lotus 1-2-3 filters include:

- [**FILTER SS TO WK1@**](#)
- [**FILTER WK3 TO SS@**](#)
- [**SS IMPORT WK3@**](#)
- [**SS IMPORT WKS@**](#)

When importing spreadsheet files which contain unsupported functions, the **Unsupported Functions** dialog box displays. There is also a **Import Problems** dialog box that will display if you have any other types of import problems such as unsupported formulas, range names and so on.

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

FILTER_XLS_TO_SS@

Converts an XLS file to an Applixware Spreadsheets file

Format FILTER_XLS_TO_SS@ (infile, outfile)

Arguments

infile	The full path name of the XLS file will be converted.
outfile	The full path name of the resulting Applixware Spreadsheets file, including the .as extension.

Description Creates an Applixware Spreadsheets from a Microsoft Excel file. This macro supports conversions from Microsoft Excel versions 3.0, 4.0, 5.0, and 7.0.

When importing spreadsheet files which contain unsupported functions, the **Unsupported Functions** dialog box displays. There is also a **Import Problems** dialog box that will display if you have any other types of import problems such as unsupported formulas, range names and so on.

For more information about Spreadsheets filters, refer to the *Spreadsheets* manual.

LOCALIZE_SS@

Creates a copy of a spreadsheet in which links are localized

Format LOCALIZE_SS@(filename, newFile)

Arguments

filename	The name, a string, of the spreadsheet document containing external links.
newFile	The name, a string, of the file in which to place the copy of the spreadsheet document. newFile should include the .as extension. If newFile does not exist, it is created. If newFile exists, the contents of the file are overwritten by the result of the copy.

Description Creates a copy of a spreadsheet file in which any external links in the original file are localized in the copy. The data that appears in the copy will be identical to the data that appears in the original, but no cells are linked to external spreadsheets in the copy.

RT_AUTOSTART@

Automatically starts and connects to the Real Time server

Format rpcChannel = RT_AUTOSTART@(serverName)

Arguments serverName The name of the server to which you are connecting. This name is the name of the executable file. For example, axmips, axtib, or triarch.

Description Starts server serverName.

RT_CONTRIB@

Publishes calculations based on Real Time data

Format RT_CONTRIB@(dstdata, newValue, gateway, service, f0, f1, ... , f17)

Arguments	dstData	A number assigned by your program that identifies the data being sent by the gateway. This is a numeric callback handle.
	newValue	[Micrognosis] The value being published. [Reuters] The name of the record being created. [Teknekron] The value or array of value being inserted (published).
	gateway	The gateway to which the data is being written.
	service	[Micrognosis] The name of the program to which you are publishing. [Reuters] The name of the service in which you are creating the record. [Teknekron] The formclass, which is the Teknekron record structure to which you are publishing data.
	f0	[Micrognosis] The name of the record you are publishing. [Reuters] The name of a pseudo-record within the service. [Teknekron] The subject, which contains the source, record, and exchange of the data you are publishing.
	f1	[Reuters] The name of a special field within the pseudo-record. This is followed by the fields containing records being created (up to 15 fields). [Teknekron] The fields containing the data being published (up to 16 fields).

Description Publishes (contribute) data to programs outside of Applixware. This ELF macro is called whenever you call the Spreadsheets RTINSERT function.

This macro must be preceded by the macro SET_MACRO_TOP_LEVEL@ in order to work correctly.

For more information, consult one of the following:

- [RTINSERT](#) (Micrognosis)
- [RTINSERT](#) (Reuters)
- [RTINSERT](#) (Teknekron)

See also [SET_MACRO_TOP_LEVEL@](#)

RT_DISABLE@

Stops all Real Time servers

Format RT_DISABLE@()

Description Stops all Real Time servers. Spreadsheets documents cannot auto-start the server when this macro is used, allowing you to work on Spreadsheets models without interference. After the servers are stopped, use [RT_ENABLE@](#) to restart the server.

When you run RT_DISABLE@, formula cells in the spreadsheet that contain an rtinsert built-in function are changed to NA. No changes to the value of the formula cells occurs until you restart Real Time, and a Spreadsheet event forces the formula cell to recalculate.

When you run RT_ENABLE@ to restart Real Time, formula cells that contain rtinsert continue to show NA until a recalculation is forced. One way to force a recalculation is to change the value of the field value cell.

See also [RT_ENABLE@](#)
[RT_LIVE_DISABLE@](#)

RT_DISCONNECT@

Disconnects Applixware process from a Real Time server

Format RT_DISCONNECT@(serverName)

Arguments serverName The name of a connected service.

Description Disconnects your Applixware process from a Real Time server by closing the channel upon which you were connected to the service. It also checks in the Real Time license when the service is down.

If a callback macro is set (using the argument to [SET_RT_CALLBACK@](#)), it will be invoked after this macro executes.

This macro is seldom used. The preferred way of disconnecting a task from a Real Time service is to run either [RT_DISABLE@](#) or [RT_STOP_ENGINE@](#).

RT_ENABLE@

Allows Spreadsheets documents to start Real Time servers

Format RT_ENABLE@()

Description Allows Spreadsheets documents to start Real Time servers. After servers are started, use **RT_DISABLE@** to stop all servers.

When you run RT_DISABLE@, formula cells in the spreadsheet that contain an rtinsert built-in function are changed to NA. No changes to the value of the formula cells occurs until you restart Real Time, and a Spreadsheet event forces the formula cell to recalculate.

When you run RT_ENABLE@ to restart Real Time, formula cells that contain rtinsert continue to show NA until a recalculation is forced. One way to force a recalculation is to change the value of the field value cell.

RT_ERROR_MSG@

Displays a Real Time error message

Format RT_ERROR_MSG@(code, string, object)

Arguments

code	A user-defined error code.
string	A user-defined error message.
obj	A user-defined Real Time object relating to the error.

Description Sends a user-defined Real Time error message to the display window.

RT_GET_CONFIG_VALUE@

Returns the value of a rtconfig variable

Format RT_GET_CONFIG_VALUE@(string)

Arguments string The name of a valid rtconfig variable.

Description Returns the value of a rtconfig variable specified in the rtconfig file.

RT_LIVE_DISABLE@

Suspends interaction with the real-time server

Format RT_LIVE_DISABLE@()

Description Suspends the current interaction with a real-time server. This macro simply tells the gateway that it should stop sending information to the task invoking the macro until it receives a RT_LIVE_ENABLE@ signal. The gateway continues to service other tasks.

Because the gateway is an independent process, it is possible to receive some data for up to a second or two after this macro executes.

See also [RT LIVE ENABLE@](#)
[RT DISABLE@](#)

RT_LIVE_ENABLE@

Restarts interaction with the real-time server

Format RT_LIVE_ENABLE@()

Description Reenables a task's interaction with a real-time server. That is, the task will resume receiving data. Only the current task is affected; that is, if other tasks had disabled the feed of data, they will remain disabled.

See also [RT LIVE DISABLE@](#)
[RT ENABLE@](#)

RT_LIVE_FEED_STATUS@

Returns TRUE if Real Time Live Feed is enabled

Format boolean = RT_LIVE_FEED_STATUS@()

Method this.ss_rt_live_feed_status@

Description Returns TRUE if a Real Time feed is enabled in the current document. Returns FALSE if no Real Time feed is enabled.

RT_REGISTER@

Registers a real-time task with the service

Format RT_REGISTER@(dstData, serverName, service, f0, f1, ..., f17)

Arguments

dstData	A number assigned by your program that identifies the data being sent by the gateway. This is a numeric callback handle. Enter 0 if you do not want to use a callback.
serverName	The name of the server gateway program.
service	[Micrognosis] The logical name of the data feed within the data distribution system; for example, MIPSPrices. [Reuters] The logical name of the date feed within the data distribution system; for example, IDN_SELECTFEED. [Teknekron] The service name should be tss.
f0	The name of the data record; for example, DEM= or JPY=. [Teknekron] The subject, which normally contains the service name, the record name, the exchange; as a reserved argument separated by periods, for example "RSF.REC.DEM=.NaE"
1. .f16	The field names; for example, BID or ASK.

Description Informs a gateway that an Applixware task wishes to receive data and which data it should send. This data will be associated with a logical feed number, dstData, created by your program.

[RT_UNREGISTER@](#)

[RT_TASK_UNREGISTER@](#)

RT_RECORD_PUBLISH@

Publishes calculations based on Real Time data

Format RT_RECORD_PUBLISH@(dstdata, servername, service, record, field_list, taskid, new_values)

Arguments

dstData	A number assigned by your program that identifies the data being sent by the gateway. This is a numeric callback handle.
servername	The gateway to which the data is being written.

service	[Micrognosis] The name of the program to which you are publishing. [Reuters] The name of the service in which you are creating the record. [Teknekron] The formclass, which is the Teknekron record structure to which you are publishing data.
f0	[Micrognosis] The name of the record you are publishing. [Reuters] The name of a pseudo-record within the service. [Teknekron] The subject, which contains the source, record, and exchange of the data you are publishing.
f1	[Reuters] The name of a special field within the pseudo-record. This is followed by the fields containing records being created (up to 15 fields). [Teknekron] The fields containing the data being published (up to 16 fields).
newValue	[Micrognosis] The value being published. [Reuters] The name of the record being created. [Teknekron]The value or array of value being inserted (published).

Description Publishes (contribute) data to programs outside of Applixware . This ELF macro is called whenever you call the Spreadsheets RTINSERT function.

This macro must be preceded by the macro SET_MACRO_TOP_LEVEL@ in order to work correctly.

For more information, consult one of the following:

- [RTINSERT](#) (Micrognosis)
- [RTINSERT](#) (Reuters)
- [RTINSERT](#) (Teknekron)

See also [SET_MACRO_TOP_LEVEL@](#)

RT_RESET_ERROR_CALLBACK@

Resets the Real Time error callback macro

Format RT_RESET_ERROR_CALLBACK@()

Description Resets the RT callback macro set by SET_RT_ERROR_CALLBACK@.

See also [SET_RT_ERROR_CALLBACK@](#)

RT_STOP_ENGINE@

Stops the Real Time task and disconnects the server channel

Format RT_STOP_ENGINE@(serverName)

Arguments serverName A Real Time server.

Description The macro kills the Real Time task and disconnects the server channel. The macro does not prevent a Spreadsheets document from auto-starting the server.

See [Stop/Restart Engine](#) for information about the Stop/Restart Engine dialog box.

RT_TASK_UNREGISTER@

Deletes a Real Time task from Applixware

Format RT_TASK_UNREGISTER@(task)

Arguments task An Applixware task id.

Description Disconnects an Applixware task permanently from all gateways. That is, the Applixware task will no longer receive data.

RT_TS_QUERY@

Format RT_TS_QUERY@(service, record, period, start_year, start_month, start_day, end_year, end_month, end_day, toss_null_points, date_order, field_names_needed)

Arguments service [Reuters] The logical name of the date feed within the data distribution system; for example, IDN_SELECTFEED.

record The name of the data record, such as "DEM="

period **d** = daily

w = weekly

m = monthly

q = quarterly

y = yearly
start_year Starting Year for the query. "96" = 1996.
start_month Starting month for the query. "1" = January
start_day Starting day for the query. "1" = 1st day of the month
end_year Ending Year for the query. "96" = 1996.
end_month Ending month for the query. "1" = January.
end_day Ending day for the query. "1" = 1st day of the month.
toss_null_points
 1 = Ignore market holiday points
 0 = Do not ignore market holiday points
date_order **DATE_ORDER#02R** = Ascending order (oldest to newest)
 DATE_ORDER#R20 = Descending order (newest to oldest)
field_names_needed
 TRUE means return field names
 FALSE means return field numbers

Description Returns a 2-dimensional array of historical information for the record specified by the record argument. The information returned pertains only to the specified period.

Since the macro returns an array, only the information in position (0,0) of the array is displayed in the cell containing the RT_TS_QUERY@ macro. The following example uses the Applixware clipboard to write the entire array to the spreadsheet.

```

MACRO ts_test
VAR array
array=RT_TS_QUERY@("IDN_SELECTFEED", "DEM=", "d", 96, 05,
                   01, 96, 06, 01, 1, true)
CLIPBOARD_PUT@(array)
SS_PASTE@("b1")
ENDMACRO
  
```

To run this macro, you must have the file triarch.prg file installed through login.am, and the axts1 executable residing in your axhome directory. See the Real Time manuals for more information.

RT_UNREGISTER@

"Unregisters" you from the real-time server process

Format RT_UNREGISTER@(service, record, dstTask, [dstData], serverName)

Arguments

service	[Micrognosis] The logical name of the data feed within the data distribution system; for example, MIPSPrices. [Reuters] The logical name of the date feed within the data distribution system; for example, IDN_SELECTFEED. [Teknekron] The subject, which normally contains the service name, the record name, the exchange, as a reserved argument separated by periods; for example "RSF.REC.DJIA.CNS"
record	The name of the data record
dstTask	The destination task id. Normally, you would use the value returned from ELF_TASK_ID@ .
dstData	A number assigned by your program that identifies the data being sent by the gateway. This is an optional numeric callback handle.
serverName	The name of the server gateway program.

Description Tells a gateway that it should stop sending the indicated data. This macro differs from RT_REGISTER@ in that you do not have to stop receiving information on all of the data that you subscribed to. In other words, if you were receiving 15 items, you could use this macro to stop receiving just three of them.

See also [RT_REGISTER@](#)

SET_RT_CALLBACK@

Sets the Real Time callback macros

Format SET_RT_CALLBACK@(genMacro[, disconnectMacro])

Arguments

genMacro	The macro invoked when data is received.
disconnectMacro	The macro invoked when a task is disconnected from the macro.

Description Names the macro that is invoked when either data is received or when the task is disconnected from the gateway.

The argument list for genMacro is the same as that for the [RT_CONTRIB@](#) macro.

SET_RT_ERROR_CALLBACK@

Sets the Real Time error callback macro

Format SET_RT_ERROR_CALLBACK@(genMacro)

Arguments genMacro The macro invoked.

Description Names the macro that is invoked when an error warning or a message is posted to the RT status dialog box.

The following argument list is passed to genMacro:

time A decimal value that corresponds to the current date and time (see [NOW@](#)).

code The error code.

obj The Real Time object relating to the error.

str The error message.

See also [RT RESET ERROR CALLBACK@](#)

spsheet_.am

```
' SS_PUTCELL@()
,
format ss_putcell_info_
    cell,          ' A1
    value,         ' num,string,formula
    nouupdate     ' if true don't update display yet !!
,
' Array index positions for SS_GET_STATUS:
,
define SSSDEX#OPEN_COL_    0
define SSSDEX#OPEN_ROW_    1
define SSSDEX#AUTO_CALC_   2
define SSSDEX#CALC_MODE_   3
define SSSDEX#EDIT_MODE_   4
define SSSDEX#POINT_MODE_  5
define SSSDEX#BOTTOM_COL_  6
define SSSDEX#BOTTOM_ROW_  7
,
format ss_status_
    open_col,
    open_row,
    auto_calc,          ' 1-auto, 0-manual
    calc_mode,          ' 0-normal, 1-row, 2-column
    edit_mode,
    point_mode,
    bottom_col,
    bottom_row,
    auto_graphing,
    calc_count,         ' 1-10 (not used for normal calc mode)
    font_size,
    print_size,
    text_align,
    num_align,
    num_style,
    num_prec,
    bold,
    face,
    underline,
    italic,
    color,
```

```

gridstyle,
wrap_text,
open_sheet,
bottom_sheet,
defcolwidth,      ' default col width
minimal_recalc,
calc_interval

,
' Calc modes for SS_GET_STATUS:
,
define SSS#NORM_CALC_ 0
define SS#ROW_CALC_ 1
define SSS#COL_CALC_ 2

,
' Format of a graph object as returned by SS_GET_GRAPH
,
format bg_graph_info_
    id,                ' numeric id of this graph
    task_id,           ' task id of graph if open (otherwise 0)
    name,              ' of doc
    type,              ' pie, bar...
    title,             ' As on graph "settings" menus
    subtitle,         ' As on graph "settings" menus
    display_title,    ' As on graph "settings" menus
    value_title,      ' As on graph "settings" menus
    group_labels,     ' As on graph "settings" menus
    display_labels,   ' As on graph "settings" menus
    data,
    legend_source,
    xygrp

format ge_graph_info_
    id,                ' numeric id of this graph
    task_id,           ' task id of graph if open (otherwise 0)
    name,              ' of doc
    type,              ' pie, bar...
    title,             ' As on graph "settings" menus
    subtitle,         ' As on graph "settings" menus
    display_title,    ' As on graph "settings" menus
    value_title,      ' As on graph "settings" menus
    group_labels,     ' As on graph "settings" menus
    display_labels,   ' As on graph "settings" menus

```

data,
unused,
xyGroups,
legend_source

format named_range_format_
name, ' name (local alias for ISR's)
range, ' location (insertion point for isrs)
docname, ' ISR - supporting doc's name
docrange, ' ISR - supporting doc's range
use_linked_attrs 'ISR - use attrs from the linked spreadsheet

format ss_style_
style,
precision,
invisible

format ss_protection_
protected,
invisible

format range_attr_
bold, ' 0 = No Bold ; 1 = Bold
italic, ' 0 = No Italic ; 1 = Italic
underline, ' 0 = No underline ; 1 = underline
dbl_underline, ' 0 = No double underline
' 1 = double underline
color, ' String name of a color, as
' displayed in the color palette of
' the Format ® Document Settings
' dialog box.
typeface, ' String name of typeface
align, ' 0 = default
' 1 = Left
' 2 = Right
' 3 = Repeat
ptsize, ' an integer indicating the point size of the
' text in the range.
wrap_text, ' 0 = No text wrap ; 1 = wrap text
valign ' 0 = default
' 1 = Top
' 2 = Center
' 3 = Bottom

```
,
```

' defines for array elements in globals for ss_get_globals() & ss_set_globals()
,

```
format ss_globals_  
  last_fill_start ,      ' Start value from Edit -> Fill  
  last_fill_incr,       ' increment value from Edit -> Fill  
  last_fill_stop,       ' stop value from Edit -> Fill  
  last_num_style        ' Last applied number style. The  
                        ' Number styles are displayed in the  
                        ' Format -> Numbers dialog box.  
  last_date_style       ' Last applied Date Style  
  last_num_prec,        ' Last applied Numeric Precision  
  current_width,        ' Current cell width. 0 indicates the default  
                        ' width  
  go_sdirection,        ' 0 = Next   1 = Previous  
  go_search,            ' Text to search for as specified on the  
                        ' Edit -> Find -> Text dialog  
  go_spl_direction,    ' 0 = Next   1 = Previous  
  go_special,           ' Cell Status as specified on the  
                        ' Edit -> Find -> Cell Status dialog  
  go_range,             ' Named Range  
  go_cell,              ' Cell Location as specified on the Edit ->  
                        ' Find -> Cell menu  
  name_change,          ' Reserved for future use  
  currtitles,           ' Last applied in the Spreadsheet  
  current_height        ' Reserved for future use
```

```
format pagebreaks_info_  
  total,  
  rows,  
  cols
```

```
format ss_line_attrs  
  style,  
  color
```

```
format ss_shade_attrs  
  style,  
  fgcolor,  
  bgcolor
```

```
format borders_info_  
  format ss_line_attrs      outline,  
  format ss_line_attrs top,
```

```

format ss_line_attrs bot,
format ss_line_attrs left,
format ss_line_attrs right,
format ss_shade_attrs shading

```

```
'color table
```

```

format ss_color@
name,
cyan,
magenta,
yellow,
black

```

```
,
```

```
' SS Object Types
```

```
,
```

```

define SSOBJ#GRAPHICS_ 0x01 /* Applixware Graphics objects */
define SSOBJ#CHART_ 0x02 /* Charts */
define SSOBJ#EQUATION 0x04 /* Applixware Equation objects */
define SSOBJ#AUDIO_ 0x08 /* Audio */
define SSOBJ#BUTTON_ 0x10 /* Buttons */
define SSOBJ#SPREADSHEET_ 0x20 /* Applixware Spreadsheet objects */
define SSOBJ#WORDS_ 0x40 /* Applixware Words objects */
define SSOBJ#DATA_ 0x80 /* Applixware Data objects */
define SSOBJ#HTML_ 0x100 /* Applixware HTML objects */
define SSOBJ#BUILDER_ 0x200 /* Applixware Builder */
define SSOBJ#UNKNOWN_ 0x400 /* Unknown Objects */
define SSOBJ#ALL 0xFFFF /* not used */

```

```
,
```

```
' Business Graph
```

```
,
```

```

format ss_graph_info_
id, ' ss internal id
task_id, ' non-zero if graph is open
name ' filename

```

```
,
```

```
' Cell/Range
```

```
,
```

```

format ss_range_
left_col,
top_row,
right_col,
bottom_row,

```

```

    start_sheet,
    end_sheet

format ss_cell_addr_
    col,
    row,
    sheet,
    abs_col,
    abs_row,
    abs_sheet

format ss_page_setup_
    width, /* width of page in inches( cms if in metric mode ) */
    height, /* height of page in inches( cms if in metric mode ) */
    lmargin, /* left margin */
    rmargin, /* right margin */
    tmargin, /* top margin */
    bmargin, /* bottom margin */
    landscape,
    center_halign,
    center_valign,
    prt_headers,
    facing_pages,
    print_to_fit,
    num_pages_wide,
    num_pages_tall,
    no_print_beyond_last_cell,
    print_gridlines /* TRUE if to print grid lines on */
    paper_type /* a number from 0 to 17 for different
                paper types
                e.g. of a paper type is "US Letter" */
,
' SS_GET_CELL@() values for the type field in ss_cell_
,
define SSC#CELL_NUM_FORMULA 100
define SSC#CELL_TEXT_FORMULA 101
define SSC#CELL_BOOL_FORMULA 102
define SSC#CELL_OBSOLETE 103
define SSC#CELL_ERROR 104
define SSC#CELL_NA 105
define SSC#CELL_PENDING 106
define SSC#CELL_CIRCULAR 107
define SSC#CELL_DIVZERO 108

```

```

define SSC#CELL_UDEFNAME      109
define SSC#CELL_NUMERR       110
define SSC#CELL_TYPERR       111
define SSC#CELL_REFERR       112
define SSC#CELL_ARGERR       113
define SSC#CELL_FAILERR      114
define SSC#CELL_DBERR1       115
define SSC#CELL_DBERR2       116
define SSC#CELL_DBERR3       117

```

```

define SSC#CELL_NUMERIC      200
define SSC#CELL_TEXT         201
define SSC#CELL_EMPTY        204
define SSC#CELL_NA_VAL       205
define SSC#CELL_ERROR_VAL    206

```

```

'cell styles - style_type field in ss_cell_
define SSC#STYLE_UNSTYLED 0
define SSC#STYLE_BOOL      1
define SSC#STYLE_GENERAL  2
define SSC#STYLE_FIXED     3
define SSC#STYLE_SCIENCE  4
define SSC#STYLE_MONEY     5
define SSC#STYLE_COMMA     6
define SSC#STYLE_PERCENT   7
define SSC#STYLE_DATE      8
define SSC#STYLE_GRAPH     9
define SSC#STYLE_TIME     10
define SSC#STYLE_DEFAULT  11
define SSC#STYLE_TEXT      15

```

```

'cell justification values - just field in ss_cell_
define SSC#JUST_LEFT      1
define SSC#JUST_RIGHT     2
define SSC#JUST_CENTER    3
define SSC#JUST_REPEAT    4

```

```

format ss_cell_
    row,          ' zero based column number, as passed
    col,          ' zero based row number, as passed
    type,         ' cell type, see below
    display_str,' as displayed in the grid
    entry_str,    ' as displayed in the editline
    style_type,   ' display style type

```

just, ' type of justification
 wrap_text, ' true if wrap_text justification is on, valid only for textcells
 protected, ' true if protected, false if not protected
 invisible, ' true if invisible, false if visible
 precision, ' extent of display style
 value, ' current value if cell is numeric
 sheet, ' zero based sheet number
 valign ' vertical alignment

format ss_chart_datum@

display_type,
 label_x_offset,
 label_y_offset,
 label_string,
 x, y, z

format ss_chart_group@

type, stack_id, custom_part_name,
 name,
 user_str, /* the group name as displayed to the user */
 x_axis,
 y_axis,
 z_axis,
 label_x_offset,
 label_y_offset,
 label_alignment,
 label_type,
 legend_string,
 series,
 main_type,
 option, 'the % explosion value for pie charts
 format arrayof gr_attribute@ attr,
 format arrayof gr_attribute@ l_attr,
 format arrayof chart_number_format@ label_format,
 format arrayof chart_datum@ data,
 hidden,
 data_range

format ss_chart_axis_info@

axis_str, 'name of axis
 axis_rngstr,
 format chart_axis_info@ axis_info, 'axis info struct
 format gr_attribute@ l_attr, 'axis label attributes
 format gr_attribute@ t_attr, 'tick label attributes

format gr_attribute@ axis_attr 'axis attributes

format ss_chart_
chart_type, ' chart type pie, bar...
series, ' all data ranges as strings !!!
axes, ' list of axes
format arrayof chart_titles@ titles,
format arrayof chart_legend@ legend,
format arrayof gr_attribute@ title_attr,
format arrayof gr_attribute@ subtitle_attr,
format arrayof gr_attribute@ footer_attr,
format arrayof gr_attribute@ leg_title_attr,
format arrayof gr_attribute@ leg_box_attr,
format arrayof gr_attribute@ leg_label_attr,
format arrayof gr_attribute@ minx_attr,
format arrayof gr_attribute@ miny_attr,
format arrayof gr_attribute@ majx_attr,
format arrayof gr_attribute@ majy_attr,
format arrayof gr_attribute@ back_attr,
format arrayof chart_decorations@ decs,
format arrayof ss_chart_group@ grp,
format arrayof ss_chart_axis_info@ x_axes,
format arrayof ss_chart_axis_info@ y_axes,
format arrayof ss_chart_axis_info@ z_axes

' Applixware Chart Info known in the Spreadsheet

format ss_chart_info_
task_id, ' task id of graph if open (otherwise 0)
name, ' of chart
main_type,
dataoffset, ' index to start of data group (after titles & axes)
format arrayof ss_chart_data,
axis_labels, ' 1st y axis labels (2.1)
legends,
orient,
path, ' Path to TempFile/External Link
extlink, ' if TRUE path is to an external link
extname, ' name of chart in external GE file
bar_overlap,
bar_margin,
groups_inited, 'true groups initialized for external charts only

```

chart_type,      ' main chart type
tmargin,        'top margin
bmargin,        'bot margin
lmargin,        'left margin
rmargin,        'right margin
null_format,    'span,grap or substitute zero for null points
gfx,            'the graphic
reformat        'true if chart needs to be reformatted

```

```

format ss_chart_attrs_
attr,           /* the actual attributes */
in_use         /* true if in use, false if not in use */

```

```

#define SS_GRAPH_NAMES 0x1 /* these match Graphic Object Types in ssdefs.h */
#define SS_CHART_NAMES 0x2
#define SS_AUDIO_NAMES 0x8
#define SS_BUTTON_NAMES 0x10

```

```

#define SS_RANGE_NAMES 0x100
#define SS_VIEW_NAMES 0x200

```

```

/* both fields == 0 gets ALL external and internal References */
#define SS_EXT_NAMES 0x400 /* if set get ONLY external References */
#define SS_INT_NAMES 0x800 /* if set get ONLY internal References */

```

```

#define FIRST_DATA_GROUP 9 ' default 1st data group
#define SS_MAIN_TITLE 0 ' match what's ssdefs.h
#define SS_SUB_TITLE 1
#define SS_FOOTER 2
#define SS_LEGEND_TITLE 3
#define SS_LEGEND 4

```

```

#define SS_X_AXIS_TITLE 5
#define SS_X_AXIS_LABELS 6

```

```

#define SS_Y_AXIS_TITLE 7
#define SS_Y_AXIS_LABELS 8

```

```

format ss_group_info_
main_type,
series,
gnames

```

```

format ss_data_range_

```

series, 'list of range vectors
legend_range, 'legend range vector
axis_range 'axis range vector

format ss_currency@
currency_str, 'the currency string itself
european, 'true if european, false if english
trailing 'true if trailing, false if leading

format ss_legend_descr@
group_name, 'the name of the data group
cell 'the cell containing the legend

format ss_name_spec@
name,
range_str

format ss_cell_attrs@
bold, /* true if bold, false if not bold, null if As Is */
italic, /* true if bold, false if not bold, null if As Is */
underline, /* 0 for none, 1 for single, 2 for double, null for As Is */
face, /* facename as a string, or null for As Is */
color, /* color */
ptsize, /* point size null or 6,8,12,14,18,24,36 */
align, /* 0 for none, 1 for left, 2 for right, 3 for center, 4 for repeat */
style,
prec

format ss_user_format@
str,
format ss_cell_attrs@ attr

format ss_object_
type,
name, ' button text
path,
x,
y,
wid,
hyt,
anchor_cell,
cellrng,
macro_to_run,
follow_move,

follow_resize,
hidden,
locked,
print,
extlink, ' if TRUE path is to an external link
task_id, ' non-zero if task is running
background ' Object is open in the background (for fast updating)

format ss_obj_info@

name,
type,
property,
hidden,
locked,
extlink,
print,
path,
macro_to_run,
no_border,
title

bitmap_path,
filter_macro

format ss_cell_loc@

column,
row,
sheet

format ss_object_loc@

format ss_cell_loc@ top_left,
format ss_cell_loc@ bot_right,
xoff1, yoff1,
xoff2, yoff2

format ss_object_pos_

x, y,
wid,
hyt

format ss_object_state_

task_id,
background,
gfx

```
format ss_current_chart@
    charts,      /* array of selected charts */
    gfx         /* the graphics pointer */
```

```
format ss_calc_options@
    mode,
    style,
    iteration_count,
    calc_interval,
    auto_chart,
    calc_background, /* this is obsolete as of rev 4.2 */
    calc_on_display, /* this is obsolete as of rev 4.2 */
    calc_rtinsert_on_display,
    calc_only_obsolete_cells,
    type_conversion,
    no_subscribe_rt_on_open,
    calc_only_cells_onscreen,
    optimal_calc      /* calculate dependent formulas */
                    /* only if cells value really changed */
    calc_before_save, 'not used
    no_register_od_on_open, ' used for stop od cells from getting registered
                        ' during file open
    no_register_rti_on_open
```

' column, row, sheet, paired-column, paired-row, paired-sheet : 0,1,2,3,4,5

```
format chart_step_order@
    ch_order,
    use_columns,
    use_rows
```

```
format chart_step_titles@
    title,
    subtitle,
    footer,
    x_axis,
    y_axis
    legend
```

' SS_GET_CELL_INFO@() is a an obsolete macro, DON'T use it. Use SS_GET_CELL@() instead
' SS_GET_CELL_INFO() array indexes

```
define SSCDEX#COL_NUM_    0 ' zero based column number, as passed
```

```

define SSCDEX#ROW_NUM_      1 ' zero based row number, as passed
define SSCDEX#CELL_TYPE_   2 ' cell type, see below
define SSCDEX#DISP_STR_    3 ' grid display string
define SSCDEX#ENTRY_STR_   4 ' entry line string
define SSCDEX#PROTECTED_   5 ' whether the cell is protected
define SSCDEX#STYLE_TYPE_  6 ' display style type
define SSCDEX#PRECISION_   7 ' extent of display style
define SSCDEX#VALUE_       8 ' current value if cell is numeric

```

```

format ss_cell_info_
    col,          ' zero based column number, as passed
    row,          ' zero based row number, as passed
    type,         ' cell type, see below
    display_str, ' grid display string
    entry_str,   ' entry line string
    protected,   ' whether the cell is protected
    style_type,  ' display style type
    precision,   ' extent of display style
    value,       ' current value if cell is numeric
sheet
,

```

```
' Cell Status codes as returned in SS_GET_CELL_INFO()
,
```

```

define SSC#CELL_IS_EMPTY_    0 ' Has no content
define SSC#CELL_IS_FORMULA_  1 ' Contains a valid formula.
define SSC#CELL_IS_CONSTANT_ 8 ' Is a constant number.
define SSC#CELL_IS_LABEL_    9 ' Is a label
define SSC#CELL_IS_OBSOLETE_ 2 ' needs to be recalculated
define SSC#CELL_IS_ERROR_    3 ' Has the ERROR value
define SSC#CELL_IS_NA_       4 ' Has the NA value
define SSC#CELL_IS_PENDING_  5 ' References an empty cell.
define SSC#CELL_IS_CIRCULAR_ 7 ' Is a member of a circularity.
define SSC#CELL_IS_STRFORMULA_ 11 'Is a string formula

```

```

format ss_view_pair@
    first, /* start of visible row/col block */
    last  /* end   of visible row/col block */

```

```

format ss_row_pair@
    row,      '0 based row number
    height    'height of the row in points */

```

```
format ss_col_pair@
```

col, '0 based column number
width 'width in characters

format ss_view_info@

sheet, '0 based sheet number
default_width, 'default column width in chars
default_height, 'default height of row in points
title_rows, 'array of title rows in this view
title_cols, 'array of title columns in this view
format arrayof ss_row_pair@ row_heights, 'array of row heights
format arrayof ss_col_pair@ col_widths, 'array of column widths
format arrayof ss_view_pair@ row_pairs,
format arrayof ss_view_pair@ col_pairs

format ss_window_info@

view_name, /* the view thats displayed in the window */
format ss_cell_loc@ top_left, /* windows top left cell */
format ss_cell_loc@ active_cell, /* windows cell cursor position */
rows, /* array of rows that are displayed in the window */
cols /* array of cols that are displayed in the window */