

The PrintGrid API

The API used in version 1 was simple but effective, given the print features supported by that version. As the original API is supported in version 2 it is documented below but programmers are advised to use the version 2 API for all new projects.

Version 1.

The PrintGrid functionality is implemented by a simple call to the relevant Public Sub on the P001 form and by supplying the required arguments thus -

```
P001.PrintGrid theGridToBePrinted, theNumberOfCopies, theTitleText, thePrintSettings
```

The print settings available were exposed by the P001 form as a Public Enum and the declaration for that Enum is shown below.

```
Public Enum PrintSettings
    SHOW_LINES = 16 'Sets Grid line Printing
    TITLE_FIRSTPAGE = 64 'Print a title on the first page
    TITLE_ALLPAGES = 128 'Print a title on all pages after first row
    REPEAT_HEADINGS = 256 'Repeat Fixed grid rows on subsequent pages
    GRID_FONT = 512 'Supports bold, italic and underlined cells in Grid
    ALLOW_SELECTED = 1024 'Allows the user to elect to print selected area of grid only
    ALLOW_DIALOGUE = 1 'Set to show the dialogue form to the user
    ALLOW_COLOURS = 2 'Allows Grid colours to be reproduced
    ALLOW_SETLINES = 4 'allows the user to select to print Grid lines
    ALLOW_TITLE = 8 'Allows the user to change the report title settings
    GRID_NORMAL = 1 + 4 + 8 + 16 + 1024 'Default settings
End Enum
```

The required print setting is a long number created by adding the values associated with each of the required components. You can just supply the long number or you can make use of the “Intellisense” feature of Visual Basic and use the PrintSettings Enum to construct the value (as shown in the code sample below).

If this version 1 call is used with version 2 of PrintGrid then the software attempts to emulate the effects of those settings as implemented in version 1. While this emulation is not perfect in every small detail it is sufficient to allow programmers to upgrade programs running version 1 of PrintGrid to version 2 without having to make any code changes. In addition, the Version 1 call is a very effective way of implementing the routine in prototypes or at early stages in an application development.

A typical program code fragment calling the version 1 API might look like this:

```
Private Sub PrintButton_Click()
    P001.PrintGrid, MsFlexGrid1, 1, “Monthly Sales Report”, PrintSettings.GRID_NORMAL
End Sub
```

Version 2. – the all new API

The revised API gives the programmer much more control over what is shown to the user and over any default values used. As this is potentially much more onerous than the version 1 API it has been arranged that if a particular setting is omitted by the programmer then a sensible default value will be assumed by the PrintGrid routines.

To use the version 2 call then you must first set the P001.GridToPrint property to point to the MsFlexGrid to be printed and you should set the P001.AllowDialogue attribute to True if you want your program user to be able to interact with the print dialogue form. Then all you have to do is call the P001.PrintGridAPI method (see the sample code below). You do not have to set any further values – they are all optional.

All of the version 2 API settings are managed through Public Properties (or attributes) of the P001 form. Most have associated Public Enums so that you can make use of the Visual Basic Intellisense feature when writing your code.

Please remember that this code is circulated as “open source” and you are free to develop, change and enhance the existing features and that includes the API. There is nothing more straightforward than changing the default values to simplify the implementation of these routines within your specific application.

Page Titles

The PrintGrid software provides facilities for printing page/report titles along with the content of your MsFlexGrid. You may set the following Public Properties associated with titles.

P001.TitlePages – set to one or more values from the TitleOption Public Enum

P001.TitleFontSize – set to a single greater than or equal to 1.0

P001.ReportTitle – set to the string value required as the title

Public Enum TitleOption

```
USER_MAY_SET = 1 'If not set then the user may not change any settings on the Title tab
NO_TITLES = 2 'No titles to be printed
FIRST_PAGE_ONLY = 4 'only the first page of the report will be titled
TITLE_ALL_PAGES = 8 'all pages will be individually titled
TITLE_ALL_ROWS = 16 'the title to be printed once on each row of pages
TITLE_FIRST_ROW_ONLY = 32 'title printed once on the top row of pages only
JUSTIFY_LEFT = 64 'the title will be left justified on the page or row
JUSTIFY_CENTRE = 128 'the title will be centre justified on the page or row
JUSTIFY_RIGHT = 256 'the title will be right justified on the page or row
REPEAT_COL_HEADINGS = 512 'fixed rows at the top of the grid will be repeated on
'subsequent pages if the report is longer than one page
FONT_BOLD = 1024 'The title will be printed using a Bold font
FONT_UNDERLINE = 2048 'the title will be printed using an underlined font
REPEAT_FIXED_COLS = 4096 'Repeats any fixed columns on all pages across a row of pages
TITLE_NORMAL = 1 + 4 + 64 + 1024 + 2048 'the default setting for the TitlePages property.
```

End Enum

The TitleFontSize property can be used to increase the relative size of the font used to print the title. A value of 1 (default if not set) would print the title using the default

font size of the MsFlexgrid (allowing for any compression selected by the user). A value of (say) 2 would print the title at twice the font size of the rest of the page.

Clearly, if the user is allowed access to the Title tab then any specific title settings may be overridden at run time but the values set by the programmer will be the default values displayed.

Print Effects

These settings and/or their default values are made by specifying a value for the P001.SetEffects property. You can use the Effects Public Enum to set this value.

Public Enum Effects

```
USER_CAN_CHANGE = 1      'If set allows the user to change these attributes
GRID_LINES = 2          'Prints the Grid Lines (default colour is Black)
GRID_COLOURS = 4       'Prints the grid lines using the relevant colours
TEXT_COLOURS = 8       'Uses the CellForeColor value for text (default Black if not set)
CELL_BACK_COLOURS = 16 'Uses the Cell (or Grid) Backcolor
COLUMN_HEAD_SHADING = 32 'shades fixed row cell backgrounds
CELL_FONT_EFFECTS = 64  'Allows CellFont settings to be printed
EFFECTS_NORMAL = 1 + 2 + 32 + 64
```

End Enum

If the P001.SetEffects property value set includes 16 (CELL_BACKCOLOURS) then 32 (COLUMN_HEAD_SHADING) is ignored and the actual grid colours are used.

Page Numbering and Margin Options

Page numbering is influenced by setting the P001.PageNumbering property and margin values may be set using the P001.TopMargin, P001.BottomMargin, P001.LeftMargin and P001.RightMargin properties.

Margins are measured from the edge of the page. Where margin values set are less than the minimum value allowed by the combination of selected printer and stationery then the minimum value will be substituted.

The margin measurement units may be set using the P001.MarginUnit property. This value may be set to any relevant Visual Basic Scalemode constant. If the value is not set then PrintGrid will assume that any margin settings are using the measurement units of the runtime PC. Programmers are advised to set this value when also setting margin values – you can not be sure of the set-up of the target PC. Setting this value does not influence the units used to indicate sheet and margin sizes on the P001 form when it is shown to the user.

The P001.PageNumbering property may be set to one or more of the values in the PageNumbers Public Enum. If this property is not set then this is the equivalent of setting the value to NUMBER_NORMAL (19).

Public Enum PageNumbers

USER_CAN_SET = 1 'If set allows the user to change page numbering and margin tab settings
AFTER_FIRST = 2 'pages are numbered after the first page
ALL_PAGES = 4 'all pages are numbered
TOP_LEFT = 8 'the page number is shown at the top left of the page
TOP_RIGHT = 16 'the page number is shown at the top right of the page
BOTTOM_LEFT = 32 'the page number is shown at the bottom left of the page
BOTTOM_CENTRE = 64 'getting the idea?
BOTTOM_RIGHT = 128
INCLUDE_PAGE_COUNT = 256 'the page number format is "Page nn of nnn"
'the default format is just "Page nn"
OVER_TOP_MARGIN = 512 'if space allows, print the page number above the top margin
UNDER_BOTTOM_MARGIN = 1024 'if space allows print the page number below the margin
TOP_CENTRE = 2048 'a late entry to the page number position
NUMBER_NORMAL = 1 + 2 + 16 'the default value for the related property.

End Enum

Printing Selected Cells

The API gives the programmer control over the dialogue option presented to the user that allows the printing of just a selected area of the grid. The SelectionPrintRule public property can be set to any value in the PrintSelect Enum.

Public Enum PrintSelect

PS_MULTICELL = 1 'Enables Print Selection if more than one cell selected (default)
PS_MULTICOL = 2 'Enables Print Selection if more than one Column is selected
PS_MULTIROW = 4 'Enables Print Selection if more than one row selected
PS_DEF_PRINT_SELECT = 8 'Change default action to print selection rather than all
PS_NOT_AVAILABLE = 16 'Disables the option to print a selected area

End Enum

While in most instances this print option will be allowed where more than one cell has been selected by the user, two alternate options exist. Programmers who set their grid SelectionMode property to flexSelectionByRow might like to consider using the PS_MULTIROW option as there will always be more than one column selected on their grids. Equally where flexSelectionByColumn has been set then the PS_MULTICOL option may be appropriate.

Setting the SelectionPrintRule property to a value that includes PS_DEF_PRINT_SELECT will ensure that the option to print just a selected area of the grid will be set by default when the dialogue is displayed. This setting will have no effect if the selected area of the grid does not meet the required selection range criteria.

The PS_NOT_AVAILABLE option can be used to disable the option to print a selected grid area.

If this property is not set then it will default to offering the user the option to print a selected area provided that more than one cell is currently selected (this is the equivalent to setting SelectionPrintRule to PS_MULTICELL)

Setting this property will have no effect if the user dialogue is not displayed. If the programmer wishes to print just a portion of the grid then several alternate methods

may be used. The rows and columns that are not to be printed could temporarily be set to heights and widths of 1. Alternately, an intermediate (non visible) MsFlexGrid could be used to pass the required cells to this print routine. This restriction is deliberate and is intended to ensure that restricted printing under program control can not be influenced by user action at run time.

Merged Cells

While the MsFlexGrid control is very effective in most circumstances the support for cell merging is idiosyncratic. Given that merged cells can be affectively applied to enhance the readability and impact of grid based output then an equally effective implementation of this characteristic was required for PrintGrid.

Please note that the following documentation applies to the PrintGrid functionality only and **differs from the documentation supplied by Microsoft** for the MsFlexGrid control.

PrintGrid checks the MergedCells attribute of the MsFlexGrid being printed. This value may be set to one of the following:

flexMergeNever (0)
flexMergeFree (1)
flexmergeRestrictRows (2)
flexMergeRestrictColumns (3)
flexMergeRestrictBoth (4)

If this value is set to 1, 2 or 4 then cell merging horizontally within rows is supported for each row that has the MergeRow() attribute set to true. If the value is set to 1, 3 or 4 then cell merging vertically within columns is supported for each column with the MergeCol() attribute set true.

Apart from setting the grid MergeCells attribute the programmer may set the P001.MergeRule property value to one or more of the constants exposed by the CellMerge Public Enum. If the MsFlexGrid is not displaying merged cells then this property value will be ignored.

```
Public Enum CellMerge
    USER_HAS_CONTROL = 1    'if set the user may change the horizontal cell merge rule
    MERGE_SAME_PAGE_ONLY = 2    'only cells on the same printed page are merged
    MERGE_ACROSS_ROWS = 4    'cells may be merged across page horizontal boundaries
    MERGE_NORMAL = 1 + 2    'the default setting of the MergeRule property.
End Enum
```

The User may be able to reset the horizontal merge cell printing behaviour at run-time if the P001.mergeRule property includes the "USER_HAS_CONTROL" value AND the user has access to the Effects tab values (please see the section on Effects above).

Merged Cells within a row.

If the relevant msFlexGrid cell merging attributes are set then PrintGrid looks for adjacent cells with the same content in each row. Cells that are also part of a merged set within a column are not included in a merged set within a row. Where a merged set of cells are identified within a row then the attributes (font, alignment etc.) of the

rightmost cell determine the display of the text within the merged cell. Sets of merged cells within a row may extend beyond the boundaries of the page being printed if the relevant P001.MergeRule property value has been set or if the user has selected that option on the Effects tab.

Merged Cells within a Column

If the relevant MsFlexGrid cell merging attributes are set then PrintGrid looks for sets of merged cells within columns. Merged cell sets are only supported within the boundaries of the printed page. In addition, cells with the same text content in a column that crosses the boundary between fixed rows and normal rows will be deemed to be separate merged sets. Within a merged set, the cell with the highest row number will influence the formatting of the text when it is printed. As PrintGrid does not support merged column sets extending beyond a printed page boundary and you can rarely be sure of the position of page breaks programmers are advised to set the cell text formatting attributes for all of the cells within a merged set to the same values.

Measurement System

The measurement units used by the PrintGrid dialogue are determined at run time by checking the set-up of the PC running the program. The currently available Windows settings are US or metric/SI. If the PC measurement system is set to the US system then measurements are presented in inches and/or decimal fractions of an inch. If the metric system is set then measurements are all presented in millimetres.

Printing Multiple Copies

You may programmatically set the P001.PrintCopies property to any positive integer value. This will set the default number of copies to be printed on the dialogue if this is displayed to the user. The value will also be passed to the selected printer driver. PrintGrid does not itself support the printing of multiple copies and this setting will only have an effect if it is supported by the relevant print driver.

Setting the Printer

If the P001 dialogue is displayed then the user may select from any printer installed on the PC running the program. The printer selection defaults to the default printer for that PC. The programmer may use the P001.SetPrinter property to define a specified printer as the default or actual printer for the output. The P001.SetPrinter property takes a variant as an argument.

Printers Collection

If the P001.SetPrinter property is set to an integer value then PrintGrid will assume this value is the position of the desired printer in the printers collection. If the number supplied is invalid then the printer will remain as the default printer set for the PC at runtime. This method of setting the printer is not recommended unless the user has selected the printer from another dialogue or related program form at run time.

Named Printer or Port (network name)

If the P001.SetPrinter property is a string then PrintGrid will attempt to match the supplied value with both the printer names and ports in the Printers collection. If a match is not found then the PC default printer is selected. The name match is case insensitive.

Programmers are advised to use the port name (e.g. \\ourprintserver\hplaserjet) when designating a specific network printer for output as this is more likely to remain fixed than the printer name on an individual PC.

Printer Orientation

You can use the PrinterOrientation property to set the default printer orientation for the dialogue or to fix the printer orientation if the dialogue is not shown. The property may be set to either of the values in the PageOrientation Enum which have the same value as the CommonDialog control Printer Orientation constants. If this value is not set then the current printer default is retained. If the user changes the printer using the PrintGrid dialogue then the orientation value will change to the default for that printer.

Counting the Output Pages

The PrintGrid routine provides facilities for counting the number of pages that would be required to print a grid without compression.

The API should be used as normal to set all of the required values that could affect the number of pages output. These specifically include the page titling and numbering options together with the grid to be measured. You can then call the CountGridPages Public Sub and read the RequestPagesHigh and RequestPagesWide properties to establish the number of pages.

A simple code sample demonstrates this feature:

```
P001.GridToPrint = MSFlexGrid1
P001.ReportTitle = "Testing My New Report"
P001.TitlePages = TitleOption.TITLE_ALL_PAGES + TitleOption.REPEAT_COL_HEADINGS
P001.SetEffects = Effects.EFFECTS_NORMAL
P001.CountGridPages
TotalPages = P001.RequestPagesHigh * P001.RequestPagesWide
```

Support for multiple grid sections

Two new measures RequestSectionsWide and RequestSectionsHigh have been added to support program control over the printing of multiple grid sections across a single physical page (please see below). After calling the CountGridPages method you can now also test the P001.RequestSectionsWide and/or P001.RequestSectionHigh properties.

If the grid extends to more than one page wide then the RequestSectionsWide value will be set to 0 (zero) – please be aware of this when including this value in any calculations. The value will otherwise be set to the largest number of sections that will be placed on any one page should this print method be selected. Again, please note,

that this is not necessarily the maximum number of grid sections that will fit on the printed page – unless the RequestSectionsHigh property is greater than 1.

The RequestSectionsHigh property will indicate the total number of pages that would be printed using the multiple grid sections method.

Please note that the RequestPagesHigh property remains at the normal value even if printing multiple grid sections on each page is set as the default behaviour (see below).

Printing Multiple Grid Sections Across a Page

These options are particularly relevant when printing very narrow (perhaps single column) grids. Where the grid is long enough to extend to multiple pages it may be possible to print two (or more) vertical sections of the grid across a single physical sheet of paper.

If it is possible to print multiple sections of a grid across the currently selected page on the currently selected printer then this option may be presented to the user on the print dialogue. The option is presented as an alternative to normal and compressed printing by displaying a third (option) radio button to the user.

The programmer has control over the default setting of the relevant Option control and over the user's ability to view and set this feature. In addition, the CountGridPages API call has been extended to allow the calling program to check if multiple grid sections could be printed (please see above) and on how many pages.

You can use the MultiColumnPrint property to control the default behaviour and user access to the facility. This property should be set to one or more values from the MultiColumn Public Enum.

Public Enum MultiColumn

MC_SET_AS_DEFAULT = 1 'Print multiple grids on the same page if they will fit

MC_DENY_MULTI_COL = 2 'do not let the user have access to the option on the dialogue

End Enum

Setting the default to print multiple columns across the page will be ignored if the grid is judged to be too large.

If this property is not set then the default will be to display the option to the user if printing multiple sections is possible.

Compressed Printing Under Program Control

If the user dialogue is displayed then print compression can be selected and controlled at run time. Facilities are also made available to support compressed printing under program control – normally when the print dialogue is not displayed to the end user.

Programmers are strongly advised to use the CountGridPages method to establish the maximum number of pages that would be printed before applying compression under program control. The values returned by the CountPages method can be influenced by

the printer page orientation and programmers are also advised to explicitly set this value or to test the results of the two alternate settings before deciding upon a suitable compression factor within their programs.

Just as in the user dialogue, the compression is set by fixing the number of pages wide and high that is required. The values are set using the RequestPagesWide and RequestPagesHigh properties. Compression is then requested by setting the SetPages property to true. This should be followed by making the normal call to the PrintGridAPI method.

If the user dialogue is displayed then the compression values set will be used as the default setting for the user dialogue. SetPages will always be re-set to False by the dialogue once the values have been initialised as compression is then under user control.

Care should be exercised when deciding upon suitable compression values. If the requested compression is considered to be too extreme (when the font size drops below 2 points) then the SetPages value will be set to False and no print output will occur. If the user dialogue is not displayed then programmers should check the PrintProgress value (please see “Checking on Print Progress” below) to ensure that their output has been printed.

The SetPages property might only be automatically re-set by making subsequent use of the CountGridPages method. Programmers are advised to set the value false after checking it upon return from the PrintGridAPI method as it is possible that it might affect the printed output from other grids within their program.

If SetPages is set True without first setting the RequestPagesWide and RequestPagesHigh properties (explicitly or by calling the CountGridPages method) then the print output will not be compressed.

Checking and Displaying Print Progress

The PrintGrid facility has an API setting (from version 2.01) that will show a print progress bar and progress description to the user at run time. If the ShowProgress public attribute is set true then the dialogue form will re-size when required and display just a progress bar and a label with the legend “Printing page nn of nn”. The progress bar and label will be updated as the routine starts to print each page.

The PrintGrid facility also provides two ways of checking on print progress or completion from the calling program.

In normal use the code executes synchronously and there are few problems likely to occur from users clicking controls during execution. There are a number of DoEvents statements in the routine however and programmers might like to be cautious and check one or both of the progress values before allowing forms to close or the grid content to change.

There are also circumstances where the code might be executed asynchronously (as part of an ActiveX DLL for instance). Then a measure of progress (or completeness)

may be more important. Programmers might consider disabling selected buttons and menu options while the print is in progress. In addition, the programmer may wish to show a suitable mouse pointer or even a progress bar to give the end user some visual feedback on the process if the built in capability to show progress is not being used.

The simplest way to check on progress is to set and check the PrintComplete property. This should be set to False before calling the PrintGrid or PrintGridAPI methods. The value will be set to True once the print has been completed. If the user dialogue is displayed and the user cancels the print then PrintComplete will again be set True.

In addition, the CountGridPages (please see “Counting the Output Pages” section above) method can be called and the total number of pages to be printed calculated from the RequestPagesHigh and RequestPagesWide properties. Calling the CountGridPages method automatically re-sets the counter of pages printed. This counter can be read using the PrintProgress property while the print process continues. The value is incremented at the start of each page.

The PrintProgress property can be used by all programmers to check that printed output has occurred after control has returned from the PrintGridAPI method.

Grid Attributes

During printing, the following MsFlexGrid attributes may be supported or may influence the output.

WordWrap
CellAlignment
BackColorFixed
ForeColorFixed
CellBackColor – Defaults to the grid BackColor
CellForeColor
GridColor
MergeCells – combined with the relevant row and/or column settings
CellFontBold
CellFontItalic
CellFontUnderline

Selected Area Printing With Column Headings

If a section of the grid greater than a single cell has been “selected” by the user then the PrintGrid dialogue (if displayed) may present the user with the option of printing the selected area only. This option is influenced by the SelectionPrintRule property – please see the relevant section above. The user may select this option and combine it with the option to “Show Column Headings On All Pages” (found on the “Title” tab) to automatically include the relevant columns from any fixed rows at the top of the grid. Program control of this feature is available by combining setting for the TitleOption property and the SelectionPrintRule property.

Selected Area Printing With Row Captions

If a section of the grid greater than a single cell has been “selected” prior to calling the PrintGrid dialogue then the user may be presented with the option of printing the

selected area only (please see the documentation for the SelectionPrintRule). The user may select this option and combine it with the option to “Show Row Captions on All Pages” (found on the Title Tab) to print the fixed columns from the left of the grid alongside their selected area. . Program control of this feature is available by combining setting for the TitleOption property and the SelectionPrintRule property.

Hidden Columns and Rows

Rows or columns with a width or height equal to (or less than) a single pixel are not printed. This provides support for hidden columns containing control information not directly relevant to the data being displayed but also allows the programmer to offer the user the option to close columns (or rows) that they would like to suppress from the printed output.

Code Sample

```
Private Sub PrintButton_Click()  
    PrintTheGrid  
End Sub
```

```
Private Sub PrintTheGrid()  
    P001.GridToPrint = MSFlexGrid1  
    P001.ReportTitle = "Testing My New Report"  
    P001.AllowDialogue = True 'let the user change some of the settings at run time  
    P001.TitlePages = TitleOption.USER_MAY_SET + TitleOption.TITLE_ALL_PAGES  
    P001.SetEffects = Effects.EFFECTS_NORMAL  
    'plus any more property settings you want to control from the calling program  
    'then just  
    P001.PrintGridAPI 'call the PrintGrid routines to display the P001 dialogue  
End Sub
```

CancelPrint

PrintGrid exposes this public method to support any Cancel Print buttons on your program forms. This method is only of relevance when the PrintGrid routine is being executed asynchronously.