

Grid Contents

Overview

Classes

Methods

Properties

Actions

Overview

A grid is a display of data in cells. Grids are drawn on an isigraph control in a form.

Some wd commands are specific to grids and have names prefixed glgrid, for example, glgridatt; see the *Window Driver Command Reference*, gl2 commands. Apart from these wd commands, the grid is implemented in J.

The code that paints the grid is in an object derived from one or more grid classes. The main grid classes included with J are:

jzgrid	basic grid class
jtgrid	table class, extends jzgrid. Displays a function table in a grid.
jwgrid	watch class, extends jzgrid. Displays a global variable in a grid.

A demo class, jfgrid (report class) is also included.

For an introduction to the grid control, see the labs *Grid Control*, *Grid Examples* and *Grid Low Level Programming*

Classes

The basic grid class is jzgrid, and this provides facilities that should be required for any grid, including:

- scrolling
- mouse events
- keyboard events
- setting attributes, e.g. font, color, border, cell size

While the jzgrid class could in theory be used directly (for a very simple grid), there would typically be another class on top of the basic grid class that provides a specific type of grid, for example the watch class jwgrid. This would include data-related definitions such as providing formatted data for the grid and support for copying and editing.

As well as the grid classes, yet another class is needed for the Windows form definition. This will include

- the isigraph control to display the grid
- code to create the grid object from the grid classes to be used
- calls to grid event handlers

If you create a new form using File|New Class, the Form Editor Wizard can be used to add the appropriate definitions.

See Watch Example for a typical class structure.

Watch Example

The following example using the watch grid illustrates the class paths:

```
dat=. ?30 30$100
```

This defines some data to be watched.

```
load 'jwatch'
```

This loads the class jwatch into the jwatch locale, which in turn loads the scripts jwgrid and jinput, each into their own locale. jwgrid in turn loads the basic grid script, jzgrid. At this stage, all that has happened is that these scripts have been loaded, and the locale namelist looks like:

```
conl ''
+-----+
|base|j|jinput|jwatch|jwgrid|jzgrid|z|
+-----+
```

No locale path settings have been made, so each locale has a path of z, as is the case with all new locales.

```
a=. conew 'jwatch'
```

This creates a new object (numbered locale) referenced by a. In a new session, it would be named '0':

```
conl ''
+-----+
|0|base|j|jinput|jwatch|jwgrid|jzgrid|z|
+-----+
```

The new locale is empty, but has a locale path that includes jwatch and z.

```
create__a 'dat'
```

create__a is called in the new locale, and gets its definition from jwatch. This verb creates a new form to display the grid, and then itself calls conew to create a jwgrid object and sets its path to include jwgrid and jzgrid:

```
grid=: ''conew'jwgrid'
```

Therefore the locale list now includes two numbered locales:

```
conl ''
+-----+
|0|1|base|j|jinput|jwatch|jwgrid|jzgrid|z|
+-----+
```

The paths in these two numbered locales are:

```
copath a          locale '0'
+-----+
```

```

|jwatch|z|
+-----+
      copath grid__a      locale '1'
+-----+-----+
|jwgrid|jzgrid|z|
+-----+-----+

```

The structure of numbered locales is returned by `costate`:

```

costate''
+-----+-----+-----+-----+
|refs|id|creator|path          |
+-----+-----+-----+-----+
|a   |0 |base   |jwatch z          |
+-----+-----+-----+-----+
|    |1 |0     |jwgrid jzgrid z |
+-----+-----+-----+-----+

```

Now suppose the mouse is clicked on one of the grid cells. The form is defined in locale `a`, so the mouse event handler `watch_grid_mbltdown` will be called there. Since this locale has no event handlers of its own, the definition in the `jwatch` locale is used. This is defined as:

```

watch_grid_mbltdown=: 3 : 'mbltdown__grid sysdata'

```

Therefore the actual handler called is `mbltdown` in the `grid` locale, which will pick its definition up from either `jwgrid` or `jzgrid`, whichever it sees first.

Methods

The following are the public methods of the base jzgrid class.

Method	Description
create	resets grid globals
destroy	destroy object (cover for codestroy)
get	get data for grid, returns 1 if new data was required
gridselect	selects grid control (cover for glsel)
init	initialize grid. Argument is: dataname name of grid data gridid id of grid control sbarid id of scrollbar sbarvid id of scrollbarv
setdata	set new data in grid
setfont0	set font 0
setfont1	set font 1
setheights	set heights for range
setWidths	set widths for range

Event handler methods

The following methods handle events for the grid control and are defined in jzgrid.

Method	Description
char	character event handler
mbldbl	mouse button left double click
mbldown	mouse button left down
mblup	mouse button left up
mbrdbl	mouse button right double click
mbrdown	mouse button right down
mbrup	mouse button right up
mmove	mouse button move
scrollbar	scroll horizontal bar
scrollbarv	scroll vertical bar
size	size event handler

Watch Methods

The following are the public methods of the watch class jwgrid. In some cases, the watch class method overrides the base method.

Method	Description
copy	copy selected cells to clipboard
create	resets grid globals
destroy	destroy method
paste	paste clipboard into grid

setdata set new data in grid
sethighlight show highlights in grid

Properties

The following are the public properties of the base jzgrid class.

Name	Default	Description
atts	see resetatts	attribute matrix
borders	{empty}	border definitions
dataname	{none}	name of variable displayed in grid
defheight	20	default cell height
defwidth	50	default cell width
editenable	0	set to 1 when editing is enabled
editflag	0	set to 1 if editing cell
extent	1 1	extent of selection
filltype	0	fill type for other cells
fixc	1	number of fixed columns
fixeddata	0	1 if all data is supplied at one go
fixr	1	number of fixed rows
fixtype	0	fill type for a fixed cell
font0	courier new 12	font0, default "'courier new" 18'
font1	arial 12	font1, default 'arial 18'
formhwnd	{none}	form window handle
gridhs	{empty}	grid heights
gridhwnd	{none}	grid control handle
gridws	{empty}	grid widths
mark	0 0	top left row,col of selected data
maxc	–	max number of columns
maxr	–	max number of rows
sbarid	{none}	id of the scrollbar
sbarvid	{none}	id of the scrollbarv
scrollc	0	cols scrolled out of view
scrollr	0	rows scrolled out of view
sizeenable	0	set to 1 if user can resize cells
skipc	0	skipped cols
skipr	0	skipped rows

The following are the public properties of the jwgrid watch class.

Name	Description
highlight	highlight expression

Actions

The following actions are defined in jzgrid.

Action	Description
arrows	move selection (scrolls to keep visible)
shift+arrows	extends range
ctrl+arrows	scrolls data
ctrl+shift+arrows	changes default cell size
home	scrolls left to col 0
ctrl+home	scrolls to row 0 and col 0
ascii characters	new data for selected cell
enter	end edit mode, move to next row
alt-enter	edit mode new line
click	selects cell
double-click	set edit mode on for current cell data, if edit is enabled
mouse-down & hold	extends range of selection to right and down
ctrl+e	enable/disable edit
ctrl+f	edit font

A specific grid may define additional or alternative behavior. For example, the watch grid adds:

ctrl+m	enter highlight expression
--------	----------------------------