Returns
none

Description

This function is quite specific in its use. Its purpose is to search for global *Glish* symbols which are either *Glish* arrays or Image `tools`, and to store them in an internal list. This list is used by other Viewer `tool` functions. After creating the updated list, it emits an event called 'datasets_updated' so that any interested parties are informed that new data might be available for their own lists. For example, the Viewerdatamanager `tool` listens for this event so that its list of data remains current. This function might also be used by a programmer after a global Image `tool` is explicitly constructed in their code.

---

viewer.widgetset - Function

Package	**display**
Module	**viewer**
Tool	**viewer**

Return the Widgetserver `tool` used by this `tool`

Synopsis
widgetset()

Returns
Widgetserver `tool`

Description

This function simply returns the Widgetserver `tool` that is used by this Viewer `tool` to build GUIs. It is generally reserved for internal use.

1.3.2 viewercanvasmanager - Tool

Package	**display**
Module	**viewer**

Tool for managing the canvases of Viewerdisplaypanel `tools`
include "viewercanvasmanager.g"

Constructors

viewercanvasmanager	Create a Viewercanvasmanager `tool`

Functions

done	Close down (destroy) this Viewercanvasmanager `tool`
getoptions	Get the current canvas settings
gui	Create a new graphical user interface for this `tool`
restoreoptions	Restore a previously saved state of this `tool`
saveoptions	Save the state of this `tool`
setoptions	Install new canvas settings
type	Return the type of this `tool`
viewer	Return the Viewer `tool` which manages this `tool`

Description

This `tool` provides services to manage and manipulate the drawing canvases of Viewerdisplaypanel `tools`. At present, this means adjusting margin spacing (to make room for axis labels, for example), selecting the number of panels for channel maps, selecting screen or paper colors (that is, white-on-black or black-on-white), and adjusting the number of colors used by colormaps on the Viewerdisplaypanel `tools`.
The graphical user interface provided by the Viewercanvasmanager `tool` is shown in Figure 1.1. A Viewercanvasmanager `tool` GUI for a Viewerdisplaypanel `tool` can be placed on the screen by selecting Canvas manager from the File menu of the Viewerdisplaypanel `tool`, or by invoking the newcanvasmanagergui function of the Viewerdisplaypanel `tool`. A single Viewercanvasmanager `tool` can have multiple GUIs on the screen at any one time, but this can become confusing to the user. The Viewercanvasmanager `tool` GUI makes use of an autogui to allow modification of the various canvas parameters.

Graphical User Interface

The graphical user interface consists of three rollups:

- Geometry This rollup controls the margins around the image. These are reserved for axis labels. If you don't use labelling you can set them to 0.
- Layout This rollup is only of interest to you if you want to create channel maps, i.e. draw frames of consecutive channels of a data cube on one display. These will be arranged from top-left to right-bottom, starting at

the frame selected in the animator. The sliders control the number of displays in x and y as well as the space between them. Note that this space is on top of the margins in Geometry. To use these appropriately set all margins to zero. This rollup won't show up if a viewercanvasmanager tool is associated with a viewerslicedp tool.

- Basic settings Basic settings control the numer of colors (index mode) and the background color of the display. The background color by default is black, but the options `Use paper colors` can be set to true so that it would look like a PS print out. The orientation of Color Wedges can also be selected here.

Canvas settings can be saved to and restored from the AIPS++ "inputs table" using the Save and Restore buttons.
The viewercanvasmanager tool is context sensitive to some degree. A Viewerdisplaypanel in RGB mode for example has a slightly different interface and on a True Color display in RGB mode you won't get an option to modify the number of colors.

viewercanvasmanager - Constructor

Package	**display**
Module	**viewer**
Tool	**viewercanvasmanager**

Create a Viewercanvasmanager `tool`

Synopsis
viewercanvasmanager(displaypanel)

Arguments

displaypanel	The Viewerdisplaypanel `tool` which will use this Viewercanvasmanager `tool` Allowed:Viewerdisplaypanel `tool`

Returns
Viewercanvasmanager `tool` or fail

Description

This is the only constructor for Viewercanvasmanager `tools`. It constructs (creates) a Viewercanvasmanager `tool` which can subsequently be used to provide canvas management to a Viewer `tool`.
This constructor should not be called by users: it is called during construction of Viewerdisplaypanel `tools`. The Viewercanvasmanager `tool` for a given Viewerdisplaypanel `tool` can be retrieved by calling its canvasmanager function.

---

viewercanvasmanager.type - Function

Package	**display**
Module	**viewer**
Tool	**viewercanvasmanager**

Return the type of this `tool`

Synopsis
type()

Returns
'viewercanvasmanager'

Description

This function simply returns the string `'viewercanvasmanager'`, which can be used to identify this `tool` as a Viewercanvasmanager `tool`. It is generally reserved for internal use.

---

viewercanvasmanager.viewer - Function

Package	**display**
Module	**viewer**
Tool	**viewercanvasmanager**

Return the Viewer `tool` which manages this `tool`

Synopsis
viewer()

Returns
Viewer `tool`

Description

This function simply returns the Viewer `tool` which owns the Viewerdisplaypanel `tool` that constructed and uses this Viewercanvasmanager `tool`. It is generally reserved for internal use.

---

viewercanvasmanager.done - Function

Package	**display**
Module	**viewer**
Tool	**viewercanvasmanager**

Close down (destroy) this Viewercanvasmanager `tool`

Synopsis
done()

Returns
T

Description

If you no longer need to use this Viewercanvasmanager `tool`, calling this function will free up its resources. Any graphical user interfacesassociated with this `tool` will be removed from the screen.
It is unwise to close down a Viewercanvasmanager `tool` by an explicit call to this function. It is generally reserved for internal use by the Viewerdisplaypanel `tool` which uses this Viewercanvasmanager `tool`.

---

viewercanvasmanager.gui - Function

Package	**display**
Module	**viewer**
Tool	**viewercanvasmanager**

Create a new graphical user interface for this `tool`

Synopsis

gui(parent, show, hasdismiss, hasdone, widgetset)

Arguments

parent	parent frame for GUI Allowed:agent or F Default: F
show	map the GUI to the screen immediately? Allowed:Boolean Default: T
hasdismiss	place a 'Dismiss' button in the GUI? Allowed:Boolean Default: F
hasdone	place a 'Done' button in the GUI? Allowed:Boolean Default: F
widgetset	Widgetserver `tool` to use for the GUI Allowed:Widgetserver `tool` Default: dws

Returns

`tool` having Viewerstandardgui interface

Description

Calling this function will create and return a new GUI which can be used to view and edit the canvas settings that are managed by this Viewercanvasmanager `tool`. By judicious use of the function arguments, the graphical user interface can optionally be placed within existing frames, and left off-screen for later use. The interface of the return value conforms to the Viewerstandardgui interface.

---

viewercanvasmanager.getoptions - Function

Package	**display**
Module	**viewer**
Tool	**viewercanvasmanager**

Get the current canvas settings

Synopsis
getoptions()

Returns
autogui full-form parameter record

Description

This function can be used to determine the current canvas settings according to this Viewercanvasmanager `tool`. The returned record is simply a full-form autogui parameter record (see the autogui documentation).
The parameters that make up the returned record are described in the documentation for the getoptions function of the Viewerdisplaypanel `tool`.

---

viewercanvasmanager.setoptions - Function

Package	**display**
Module	**viewer**
Tool	**viewercanvasmanager**

Install new canvas settings

Synopsis
setoptions(options)

Arguments

options	autogui short-form parameter record describing the requested state Allowed:record

Returns
T

Description

This function should be used to set the state of the canvas managed by this `tool`, according to the field names of the record returned by the getoptions `tool function`. Installing new options on a Viewercanvasmanager `tool` will generally cause the refresh (that is, a total re-drawing) of the Viewerdisplaypanel `tool` which it manages.

---

viewercanvasmanager.saveoptions - Function

Package	**display**
Module	**viewer**
Tool	**viewercanvasmanager**

Save the state of this `tool`

Synopsis
saveoptions(setname)

Arguments

setname	name under which to store this state Allowed:string

Returns
T (success) or F (failure)

Description

This function saves the current state (options) of this Viewercanvasmanager `tool` in the AIPS++ `Inputs Table`. The options are stored as inputs for the Viewercanvasmanager `tool`, with the "method name" set to the `setname` argument given in the call to this function.

Example
An example should show how this can be used:

```
- mv := viewer();
- mdp := mv.newdisplaypanel();
- mcm := mdp.canvasmanager();
- mcm.setoptions([leftmarginspacepg=0]);
- mcm.saveoptions('my canvas options');
```

---

viewercanvasmanager.restoreoptions - Function