STACK OBJECTS

1/ FESTIVAL_STACK OBJECT

1.1/ DESCRIPTION

The FESTIVAL_STACK object is a generic object for the implementation of a push-pull stack mechanism.

1.2/ SUPERCLASSES

IDL_CONTAINER

1.3/ PROPERTIES

1.4/ METHODS

1.4.1/ FESTIVAL_STACK::INIT

Syntax

Result = Obj->[FESTIVAL_STACK::]Init ([, CAP_MAX=capMax])

Description :

The FESTIVAL_STACK::INIT function method constructs a FESTIVAL_STACK object, with a specified capacity given in megabytes.

Return Value :

The FESTIVAL_STACK::INIT function method returns 1 if initialization was successful, 0 otherwise.

Arguments :

None

Keywords :

CAP_MAX

Capacity of the stack in Megabytes. Defualt is 300Mb.

1.4.2 FESTIVAL_STACK::PUSH

Description :

The FESTIVAL_STACK::PUSH procedure method adds an object reference to the stack, if that object reference doesn't already exist in the stack. When the maximum capacity is reached, the oldest object references are pulled out of the stack prior to pushing the new object reference.

Syntax :

obj->[FESTIVAL_STACK::]Push, objectRef

Arguments :

objectRef

An object reference be added to the stack

Keyword :

None

1.4.3/ FESTIVAL_STACK::PULL

The STACK::PULL procedure method removes an object reference from the stack. The memory associated with the pulled object is not free.

Syntax :

Obj->[FESTIVAL_STACK::]Pull, objectRef

Arguments :

objectRef

A valid object reference to be pulled out of the stack.

Keywords :

None.

1.4.4/ FESTIVAL_STACK::CLEANUP

Syntax :

OBJ_DESTROY, Obj

or

Obj->[STACK::]Cleanup (In a lifecycle method only.)

Description :

The STACK::CLEANUP procedure method performs all cleanup on the object.

Arguments :

None

Keywords :

None

2/ FESTIVAL_HISTORY_STACK OBJECT

2.1/ DESCRIPTION

- The FESTIVAL_HISTORY_STACK object is used to store FESTIVAL_IMAGE objects (cf. 2/).
- It allows to access rapidly an image without recalculation, if that image is available in the stack, or to access an image usable for the current calculation.
- It has a maximum capacity expressed in Megabytes.
- When the maximum capacity is reached, some FESTIVAL_IMAGE object references are pulled

out of the stack, and associated memory is free.

2.2/ SUPERCLASSES

FESTIVAL_STACK

2.3/ PROPERTIES

2.4/ METHODS

2.4.1/ FESTIVAL_HISTORY_STACK::INIT

Syntax

Result = Obj->[FESTIVAL_HISTORY_STACK::]Init ([, CAP_MAX=capMax])

Description :

The FESTIVAL_HISTORY_STACK::INIT function method constructs a FESTIVAL_HISTORY_STACK object, with a specified capacity given in megabytes.

Return Value :

The FESTIVAL_HISTORY_STACK::INIT function method returns 1 if initialization was successful, 0 otherwise.

Arguments :

None

Keywords :

CAP_MAX

Capacity of the stack in Megabytes.

2.4.2 FESTIVAL_HISTORY_STACK::PUSH

Description :

The FESTIVAL_HISTORY_STACK::PUSH procedure method adds an object reference to the history stack, if that object reference doesn't already exist in the stack. When the maximum capacity is reached, some object references are pulled out of the stack, the associated memory is freed, prior to the pushing of the new object reference.

Syntax :

obj->[FESTIVAL_HISTORY_STACK::]Push, objectRef

Arguments :

objectRef

An object instance be added to the history stack

Keyword :

None

2.4.3/ FESTIVAL_HISTORY_STACK::PULL

The STACK::PULL procedure method removes an object reference from the stack. The memory associated with the pulled object is not free.

Syntax :

Obj->[FESTIVAL_HISTORY_STACK::]Pull, objectRef

Arguments :

objectRef

A valid object reference to be pulled out of the stack.

Keywords :

None.

2.4.4/ FESTIVAL_HISTORY_STACK::SEARCH

The FESTIVAL_HISTORY_STACK::SEARCH function returns an object corresponding to some search criteria.

Syntax :

Result = Obj->[FESTIVAL_HISTORY_STACK::]Search([,PROBE=probe][,INSTRUMENT=instrument][,IMAGE_INDEX=imageIndex][,INSTRUMENT=instrument][,FILTER=filter][,PROJECTION_NAME=projectionName][,PROJECTION_CENTER=projectionCenter][,CALIBRATION=calibration][,COORDINATE_SYSTEM=coorSystem][,SCIENTIFIC_UNFILTERED=scientificUnfiltered][,SCIENTIFIC_FILTERED_BADPROJ=scientificFilteredBadProj])

Return Value :

The FESTIVAL_HISTORY_STACK::SEARCH function method returns an object references array corresponding to the request. Default is NULL object.

Arguments :

Keywords :

PROBE

A case insensitive string giving the probe name. Possible values are “SOHO”, “STEREOA”, “STEREOB”.

INSTRUMENT

A case insensitive string giving the name of the instrument. Possible values are “EIT”, “C1”, “C2”, “C3”, “EUVI”, “HI1”, “HI2”, “COR1”, “COR2”.

IMAGE_INDEX

Integer value giving the relative image index.

FILTER

A case insensitive string array giving the image filtering sequence. Default is “None”.

PROJECTION_NAME

A case insensitive string giving the name of the projection used for that image. Default is “None”.

PROJECTION_CENTER

A 2 elements array giving the (latitude, longitude) coordinates of the projection center (e.g. Sun center). Default is [0, 0].

CALIBRATION

A case insensitive string giving the name of the calibration algorithm used for that image. Default is “None”.

COORDINATE_SYSTEM

A case insensitive sting giving the name of the coordinate system used for that image. Default is “None”.

SCIENTIFIC_UNFILTERED

Set this keyword to a name variable that will return an object. containing the required scientific unfiltered buffer (e.g. Right calibration AND filering sequence = “None”).

SCIENTIFIC_FILETERED_BAD_PROJ

Set this keyword to a name variable that will return an object containing the required scientific filtered buffer, but another projection than the required.

2.4.5/ FESTIVAL_HISTORY_STACK::CLEANUP

Syntax :

OBJ_DESTROY, Obj

or

Obj->[STACK::]Cleanup (In a lifecycle method only.)

Description :

The STACK::CLEANUP procedure method performs all cleanup on the object.

Arguments :

None

Keywords :

None

3/ FESTIVAL_VISUALISATION_STACK OBJECT

3.1/ DESCRIPTION

The FESTIVAL_VISUALISATION_STACK object is dedicated to the management of the graphics objects in the data visualization GUI. It contains a number of FESTIVAL_IMAGE objects and any number of IDL graphics atoms (IDLGRPOLYGON, IDLGRPOLYLINE, ...).

- **It allows to add a visualisable object into the visualization at a given position.**
- A visualisable object can be a FESTIVAL_IMAGE object or any IDL graphics atoms

(IDLGRPOLYGON, IDLGRPOLYLINE, ...).

- **Remove a visualisable object from the visualization.**

3.2/ SUPERCLASSES

IDLGRMODEL

3.3/ PROPERTIES

3.4/ METHODS

3.4.1/ FESTIVAL_VISUALISATION_STACK::INIT

Syntax

Result = Obj->[FESTIVAL_VISUALISATION_STACK::]Init ()

Description :

The FFESTIVAL_VISUALISATION_STACK ::INIT function method constructs a FESTIVAL_VISUALISATION_STACK object.

Return Value :

The FESTIVAL_VISUALISATION_STACK::INIT function method returns 1 if initialization was successful, 0 otherwise.

Arguments :

None

Keywords :

None

3.4.2/ FESTIVAL_VISUALISATION_STACK::PUSH

Description :

The FESTIVAL_VISUALISATION_STACK::PUSH procedure method adds a valid object reference to the visualization stack, if that object reference doesn't already exist in the stack. Any FESTIVAL_IMAGE can be added to the visualization stack, at a position defined by its visualization order. Any IDL graphics object can be added at position 0 (e.g. on the foreground).

When a FESTIVAL_IMAGE object reference is added (associated to a specific instrument), the previous FESTIVAL_IMAGE object reference for that instrument is pulled out of the visualization stack.

Syntax :

obj->[FESTIVAL_VISUALISATION_STACK::]Push, objectRef

Arguments :

objectRef

An object reference to be added to the visualization stack.

Keyword :

None

3.4.3/ FESTIVAL_VISUALISATION_STACK::PULL

Description :

The FESTIVAL_VISUALISATION_STACK::PULL procedure method removes a valid image object reference from the visualization stack.

Syntax :

obj->[FESTIVAL_VISUALISATION_STACK::]Pull, objectRef

Arguments :

objectRef

An object reference to be removed from the visualization stack..

Keyword :

None