

## GUI\_PatternDefnDelete – Pattern Definition Delete

### Revision history

2017-03-19	Robin Lamacraft	Original draft
2018-06-09	Rod Thompson	Replace GUI_PanelConfigEdit with GUI_Select Configuration Replace 'screen' with 'window' Replace BR_PanelConfig with BR_WindowConfig

### SCOPE

This GUI module displays a window to delete a Pattern Definition instance. The detail depends on the Pattern Purpose of the currently selected Pattern Definition instance. The Pattern Definitions are aligned with the Node Entities within HRE. The actions from this window do not extend beyond ensuring that no data depends on the existence of this Pattern Definition instance. In particular, it will detect whether another object instance will become isolated if this Pattern Definition instance is deleted.

### Scope Variations:

Node Entity Type	Node Entity Differences
ITEM	(hidden)
→ External File	
→ Image	
→ Internal Text	
→ Another Item Type	(like Artworks)
→ Biological	(hidden)
→→ Person	
→→ Another Bio Type	(like Plants)
GROUP (hidden)	(hidden)
→ Event	
→ Task	
→ Location	
→ Repository	
→ Source	
→ Another Group Type	(like Organizations)

### LOOK AND FEEL

The window has 2 horizontal parts:

- The heading section shows
  - the focus Pattern Definition instance HRE-ID
  - the name of the Pattern Definition instance (if it has one)
  - the Node Entity Type.
- A collection of buttons including “Configure”, “Delete”, “Apply”, “Cancel” and “Output”:
  - “Configure” opens a window that allows the creation and re-use of window content and layout
  - “Delete” starts a search to check that it is possible to delete Pattern Definition without causing inconsistency in the database:
    - This search lists on the window the other objects that require the Pattern Definition to continue to exist.
    - This scrollable table has columns HRE-Id, Object Type, Object Name (if it has one)
  - This search has 2 possible outcomes:

- “Apply” is enabled on clicking if no dependent objects were found. “Apply” removes the Pattern
- “Output” - If there are other objects that require the Pattern to continue to exist, then “Output” can save this list for analysis as a file or to print it. There may be several reasons the Pattern can’t be removed
- “Cancel” does not delete the Pattern.

[Needs a mockup diagram here]

## ACTIONS

The fundamental operations are:

1. Open the window according to its saved Window Layout (BR\_WindowConfig)
2. Populate the property editor pane with values for the selected object
3. Populate the heading
4. Perform the search
5. Delete the Entity
6. Output the result of the search.

## USED BY:

1. All links (not objects) that have “Pattern Definition Edit” entries in the menu use this GUI\_PatternDefnDelete as their dependent coding
2. Almost any link type, either project-oriented or application-oriented has a GUI\_PatternDefnDelete variant. Because these GUI elements create mouse and keyboard events, each of these GUI windows must have unique identities. This means that the basic window layout can be defined as an abstract class where each separate real class contains the object type specific code listening for its specific events to act upon.

## DATA CONTROLLED BY THIS MODULE

1. None.

## REQUIRED DATA CONTROLLED BY OTHER MODULES

1. HRE-ID.

## REQUIRED SERVICES

1. GUI\_Select Configuration
2. GUI\_FieldTranslationEdit
3. GUI\_Output
4. BR\_Setting
5. BR\_WindowConfig
6. BR\_Pattern
7. BR\_EntityLink.

## APPLICATION PROGRAMMING INTERFACE (API)

1. Need Details.

## EVENT ACTIONS

1. Need details of event (keyboard or mouse) and the description of the action.

## WARNING CONDITIONS

1. Need details of the condition that raised the warning, example message and possible next steps.

## **ERROR CONDITIONS**

1. Need to record the condition that raised the error, example message and possible next steps.