GUI_TagDelete – Tag Delete

Robin Lamacraft 2017-03-19

SCOPE

This GUI module displays a screen to delete a Tag instance. The detail depends on the Tag Purpose of the currently selected Tag instance. The actions from this screen do not extend beyond ensuring that no data depends on the existence of this Tag instance. In particular, it will detect whether another object instance will become isolated if this Tag instance is deleted.

Scope Variations:

Tag Purpose	Tag Purpose Differences
Associate	Delete the selected Associate Tag instance between an Entity Node instance and an Event or Task instance.
Heading	Delete the selected Heading Tag instance from its parent Node Entity instance.
Member	Delete the selected Member Tag instance from the linked Group Node Entity instance.
Name	Delete the selected Name Tag instance from its parent Node Entity instance.
Related	Delete the selected Related Tag instance between the 2 Node Entity instances.

LOOK AND FEEL

The screen has the following horizontal parts:

- The heading section shows
  - the focus Tag instance HRE-ID
  - the name of the Tag instance (if it has one)
  - the Tag Purpose.

Heading Variations:

Tag Purpose	Tag Purpose Differences
Associate	Also shows: • The Visible ID and Primary or Selected Name of the Associated Node Entity instance (if it has a name) • The Visible ID and Primary or Selected Name of the associated Event or Task Node Entity instance (if it has a name).
Heading	Also shows: • The Visible ID and Primary or Selected Name of the parent Node Entity instance (if it has a name).
Member	Also shows: • The Visible ID and Primary or Selected Name of the Group Node Entity instance (if it has a name).
Name	Also shows: • The Visible ID and Primary or Selected Name of the parent Node Entity instance (if it has a name).
Related	Also shows: • The Visible ID and Primary or Selected Name of both Node Entity instances (if they have a name).

- A collection of buttons including “Configure”, “Delete”, “Apply”, “Cancel” and “Output”:
  - “Configure” opens a screen that allows the creation and re-use of screen content and layout

- "Delete" starts a search to check that it is possible to delete the Tag without causing inconsistency in the database:
  - This search lists on the screen the other objects that require the Tag to continue to exist
  - This scrollable table has columns of HRE-Id, Object Type, Object Name (if it has one).

Search Variations:

Tag Type	Tag Type Differences
Associate	The Associate search must check that other Tag Associates with the same Role have the plurality of that Role adjusted if the current Associate Tag instance is deleted.
Heading	There is no need for a Heading dependency search in this case.
Member	There is no need for a Member dependency search in this case.
Name	There is no need for a Name dependency search in this case. A primary Name can't be deleted until it has been marked as not-Primary.
Related	There is no need for a Related dependency search in this case. However, if the relationship is marked Primary it can't be deleted until it has been marked as not-Primary.

- This search has 2 possible outcomes:
  - "Apply" is enabled if no dependent objects were found. Clicking "Apply" removes the Tag
  - "Output" - If there are other objects that require the Tag 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 Tag can't be removed
  - "Cancel" does not delete the Tag.

[Needs a mockup diagram here]

ACTIONS

The fundamental operations are:

1. Open Screen according to its saved Screen Layout (BR_PanelConfig)
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 "Edit" entries in the menu use this GUI_TagDelete as their dependent coding
2. Almost any link type, either project-oriented or application-oriented has a GUI_TagDelete variant. Because these GUI elements create mouse and keyboard events each of these GUI screens must have unique identities. This means that the basic screen layout can be defined as an abstract class where each separate real class contains the object type specific code that is listened for, to act on its specific events .

DATA CONTROLLED BY THIS MODULE

1. None.

REQUIRED DATA CONTROLLED BY OTHER MODULES

1. HRE-ID.

REQUIRED SERVICES

1. GUI_PanelConfigEdit
2. GUI_FieldTranslationEdit
3. GUI_Output
4. BR_Setting
5. BR_PanelConfig
6. BR_Tag
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.