

# SAME-AS

Specifies an equivalence relationship between two controls ("objects"), such that both objects are considered to represent the same user action ("command"). This allows Natural to ensure the coherency of the user interface implicitly. Coherency implies that all user interface elements representing the same command have the same state at all times. For example, if a menu item and a tool bar item represent the same command, then, at any one time, either both should be enabled, or both disabled. Furthermore, Natural ensures that the same event-handling code is executed when a command is invoked, regardless of which menu or tool bar item invoked it.

The referenced object may be a signal or a menu item. The referencing object may be a tool bar item or a menu item. A SAME-AS relationship between two menu items is not allowed. A two-step approach, whereby a tool bar item references a menu item, which itself references a signal, is permitted, but not recommended. Instead, the tool bar item should, like the menu item, reference the signal directly. Note that a SAME-AS relationship between objects belonging to different dialogs is permitted. Thus, objects in MDI child dialogs may reference objects in the MDI frame dialog.

For each command, the SAME-AS relationships form a dependency tree. The object within the tree that does not have a SAME-AS relationship to another object is the *root* object. Any object in the tree to which one or more SAME-AS relationships exist is referred to as a *referenced* object, and the objects which (directly or indirectly) reference it are referred to as that object's *dependent* objects.

- ACCELERATOR
- BACKGROUND-COLOUR-NAME
- BACKGROUND-COLOUR-VALUE
- BITMAP-FILE-NAME
- CHECKED
- DIL-TEXT
- ENABLED
- MENU-ITEM-TYPE
- SHARED
- TOOLTIP
- VISIBLE

Propagation implies that the same attribute is implicitly set to the same value for each object to which the attribute setting is propagated. Furthermore, when a SAME-AS relationship is set to any value other than NULL-HANDLE, the values of all the above attributes are immediately copied from the referenced object to the referencing object. Setting the SAME-AS attribute to NULL-HANDLE breaks the relationship and restores the object's creation-time attribute values.

In addition, a click received for a non-root object results in a **CLICK** event being raised for the root object instead. This ensures that the same code is executed regardless of how the command is accessed.

As an example, suppose a command may be accessed via a menu bar menu item, a context menu item, or a tool bar item. Instead of maintaining a set of state attributes for each of these three objects, it is easier to create a signal to represent the command, and to set the **SAME-AS** attribute of all three objects to the handle of this signal. Now, only the signal's attributes need to be maintained by the program. For example, setting the **CHECKED** attribute of the signal to **TRUE** implicitly also sets the **CHECKED** attribute of both menu items and the toolbar item to **TRUE**, thus checking them. Furthermore, regardless of which one of the menu or tool bar items the user clicks, a **CLICK** event is received for the signal instead.

<b>Applies to</b>	Menu item, tool bar item.
<b>Data Type</b>	HANDLE
<b>Default Value</b>	NULL-HANDLE
<b>Possible Values</b>	NULL-HANDLE / any signal or (for tool bar items only) menu item handle