DefaultFont

DefaultFont [**/U**] "***fontName***"

The DefaultFont operation sets the default font to be used in graphs for axis labels, tick mark labels and annotations, and in page layouts for annotations.

Parameters

"fontName" should be a font name, optionally in quotes. The quotes are not required if the font name is one word.

Flags

/U	Updates existing graphs and page layouts immediately to use the new default font.

DefaultGUIControls

DefaultGUIControls [**/Mac/W=*winName*/Win**] [***appearance***]

The DefaultGUIControls operation changes the appearance of user-defined controls.

Note: The recommended way to change the appearance of user-defined controls is to use the Miscellaneous Settings dialog's Native GUI Appearance for Controls checkbox in the Compatibility tab, which is equivalent to `DefaultGUIControls native` when checked, and to `DefaultGUIControls os9` when unchecked.

Use `DefaultGUIControls/W=`*winName* to override that setting for individual windows.

Parameters

appearance may be one of the following:

native	Creates standard-looking controls for the current computer platform. This is the default value.
os9	Igor Pro 5 appearance (quasi-Macintosh OS 9 controls that look the same on Macintosh and Windows).
default	Inherits the window appearance from either a parent window or the experiment-wide default (only valid with /W).

Flags

/Mac	Changes the appearance of controls only on Macintosh, and it affects the experiment whenever it is used on Macintosh.
/W=*winName*	Affects the named window or subwindow. When omitted, sets an experiment-wide default. When identifying a subwindow with *winName*, see **Subwindow Syntax** on page III-92 for details on forming the window hierarchy.
/Win	Changes the appearance of controls only on Windows, and it affects the experiment whenever it is used on Windows.

Details

If *appearance* is not specified, nothing is changed. The current value for appearance is returned in S_value.

If *appearance* is specified the previous appearance value for the window- or experiment-wide default is returned in S_value.

With /W, the control appearance applies only to the specified window (Graph or Panel). If it is not used, then the settings are global to experiments on the current computer. **Tip**: Use `/W=#` to refer to the current active subwindow.

The /Mac and /Win flags specify the affected computer platform. If the current platform other than specified, then the settings are not used, but (if native or OS9) are remembered for use in window recreation macros or experiment recreation. This means you can create an experiment that with different appearances depending on the current platform.

If neither /Mac nor /Win are used, it is implied by the current platform. To set native appearance on both platforms, use two commands:

```
DefaultGUIControls/W=Panel0/Mac native
DefaultGUIControls/W=Panel0/Win native
```

Note: The setting for DefaultGUIControls without /W is not stored in the experiment file; it is a user preference set by the Miscellaneous Settings dialog's Native GUI Appearance for Controls checkbox in the Compatibility tab. If you use DefaultGUIControls native or DefaultGUIControls os9 commands, the checkbox will not show the current state of the experiment-wide setting. Clicking Save Settings in the Miscellaneous Settings dialog will overwrite the DefaultGUIControls setting (but not the per-window settings).

In addition to the experiment-wide appearance setting and the window-specific appearance setting, an individual control's appearance can be set with the appropriate control command's appearance keyword (or a ModifyControl appearance keyword). A control-specific appearance setting overrides a window-specific appearance, which in turn overrides the experiment-wide appearance setting.

Although meant to be used before controls are created, calling DefaultGUIControls will update all open windows.

DefaultGUIControls does not change control fonts or font sizes, which means you can create controls that look "native-ish" without having to readjust their positions to avoid avoid shifting or overlap. However, the smooth font rendering that the Native GUI uses on Macintosh does change the length of text slightly, so some shifting will occur that affects mostly controls that were aligned on their right sides.

The native appearance affects the way that controls are drawn in **TabControl** and **GroupBox** controls.

TabControl Background Details

Unlike the os9 appearance which draws only an outline to define the tab region (leaving the center alone) the native tab appearance fills the tab region. Fortunately, TabControls are drawn before all other kinds of controls which allows enclosed controls to be drawn on top of a tab control regardless of the order in which the buttons are defined in the window recreation macro.

However the drawing order of native TabControls does matter: the top-most TabControls draws over other TabControls. (The top-most TabControl is listed last in the window recreation macro.) The os9 appearance allows a smaller (nested) TabControl to be underneath the later (enclosing) TabControl because tabs normally aren't filled. Converting these tabs to native appearance will cause nested tab to be hidden.

To fix the drawing order problem in an existing panel, turn on the drawing tools, select the arrow tool, right-click the enclosing TabControl, and choose Send to Back to correct this situation. If the TabControl itself is inside another TabControl, select that enclosing TabControl and also choose Send to Back, etc.

To fix the window recreation macro or function that created the panel, arrange the enclosing TabControl commands to execute before the commands that create the enclosed TabControls.

A natively-drawn TabControl draws any drawing objects that are entirely enclosed by the tab region so that it behaves the same as an os9 unfilled TabControl with drawing objects inside.

Groupbox Control Background Details

GroupBox controls, unlike TabControls, are not drawn before all other controls, so the drawing order always matters if the GroupBox specifies a background (fill) color and it contains other controls.

You may find that enabling native appearance hides some controls inside the GroupBox. They are probably underneath (before) the GroupBox in the drawing order.

To fix this in an existing panel, turn on the drawing tools, right-click on the GroupBox and choose Send to Back. To fix the window recreation macro or function that created the panel, arrange the GroupBox commands to execute before the commands that create the enclosed controls.

A natively-drawn GroupBox draws any drawing objects that are entirely enclosed by the box; an os9 filled GroupBox does not.

See Also

The **DefaultGUIFont**, **ModifyControl**, **Button**, **GroupBox**, and **TabControl** operations.

Chapter III-14, **Controls and Control Panels**, for details about control panels and controls.

Control Panel Units on page III-444 for a discussion of the units used for controls.