G2 Dialog Utility

User's Guide

Version 2.3 Rev. 0

G2 Dialog Utility User's Guide, Version 2.3 Rev. 0
May 2007

The information in this publication is subject to change without notice and does not represent a commitment by Gensym Corporation.

Although this software has been extensively tested, Gensym cannot guarantee error-free performance in all applications. Accordingly, use of the software is at the customer's sole risk.

Gensym Corporation
52 Second Avenue
Burlington, MA 01803 USA
Telephone: (781) 265-7100
Fax: (781) 265-7101

Part Number: DOC004-230

Contents Summary

Preface

Describes this guide and the conventions that it uses.

About this Guide

This guide describes the G2 Dialog Utility (GDU) module. This module extends the custom standard dialog functionality that G2 provides to enable the rapid building and deployment of native Windows dialogs.

Audience

This guide is for G2 developers who want to customize applications, using a set of standard application programmers' interface (API) procedures and methods, and built-in classes. It assumes familiarity with the G2 procedure language.

Conventions

This guide uses the following typographic conventions and conventions for defining system procedures.

Typographic

Convention Examples	Description
g2-window, g2-window-1, ws-top-level, sys-mod	User-defined and system-defined G2 class names, instance names, workspace names, and module names
history-keeping-spec, temperature	User-defined and system-defined G2 attribute names
true, 1.234, ok, “Burlington, MA”	G2 attribute values and values specified or viewed through dialogs
Main Menu > Start KB Workspace > New Object create subworkspace Start Procedure	G2 menu choices and button labels
conclude that the x of y ...	Text of G2 procedures, methods, functions, formulas, and expressions
new-argument	User-specified values in syntax descriptions
text-string	Return values of G2 procedures and methods in syntax descriptions
File Name, OK, Apply, Cancel, General, Edit Scroll Area	GUIDE and native dialog fields, button labels, tabs, and titles
File > Save Properties	GMS and native menu choices
workspace	Glossary terms

Convention Examples	Description
`c:\Program Files\Gensym\`	Windows pathnames
`/usr/gensym/g2/kbs`	UNIX pathnames
`spreadsh.kb`	File names
`g2 -kb top.kb`	Operating system commands
`public void main()` `gsi_start`	Java, C and all other external code

Note Syntax conventions are fully described in the *G2 Reference Manual*.

Procedure Signatures

A procedure signature is a complete syntactic summary of a procedure or method. A procedure signature shows values supplied by the user in *italics*, and the value (if any) returned by the procedure *underlined*. Each value is followed by its type:

g2-clone-and-transfer-objects
(*list*: class item-list, *to-workspace*: class kb-workspace,
delta-x: integer, *delta-y*: integer)
–> *transferred-items*: g2-list

Related Documentation

G2 Core Technology

- *G2 Bundle Release Notes*
- *Getting Started with G2 Tutorials*
- *G2 Reference Manual*
- *G2 Language Reference Card*
- *G2 Developer's Guide*
- *G2 System Procedures Reference Manual*

- *G2 System Procedures Reference Card*
- *G2 Class Reference Manual*
- *Telewindows User's Guide*
- *G2 Gateway Bridge Developer's Guide*

G2 Utilities

- *G2 ProTools User's Guide*
- *G2 Foundation Resources User's Guide*
- *G2 Menu System User's Guide*
- *G2 XL Spreadsheet User's Guide*
- *G2 Dynamic Displays User's Guide*
- *G2 Developer's Interface User's Guide*
- *G2 OnLine Documentation Developer's Guide*
- *G2 OnLine Documentation User's Guide*
- *G2 GUIDE User's Guide*
- *G2 GUIDE/UIL Procedures Reference Manual*

G2 Developers' Utilities

- *Business Process Management System User's Guide*
- *Business Rules Management System User's Guide*
- *G2 Reporting Engine User's Guide*
- *G2 Web User's Guide*
- *G2 Event and Data Processing User's Guide*
- *G2 Run-Time Library User's Guide*
- *G2 Event Manager User's Guide*
- *G2 Dialog Utility User's Guide*
- *G2 Data Source Manager User's Guide*
- *G2 Data Point Manager User's Guide*
- *G2 Engineering Unit Conversion User's Guide*
- *G2 Error Handling Foundation User's Guide*
- *G2 Relation Browser User's Guide*

Bridges and External Systems

- *G2 ActiveXLink User's Guide*
- *G2 CORBALink User's Guide*
- *G2 Database Bridge User's Guide*
- *G2-ODBC Bridge Release Notes*
- *G2-Oracle Bridge Release Notes*
- *G2-Sybase Bridge Release Notes*
- *G2 JMail Bridge User's Guide*
- *G2 Java Socket Manager User's Guide*
- *G2 JMSLink User's Guide*
- *G2-OPC Client Bridge User's Guide*
- *G2 PI Bridge User's Guide*
- *G2-SNMP Bridge User's Guide*
- *G2-HLA Bridge User's Guide*
- *G2 WebLink User's Guide*

G2 JavaLink

- *G2 JavaLink User's Guide*
- *G2 DownloadInterfaces User's Guide*
- *G2 Bean Builder User's Guide*

G2 Diagnostic Assistant

- *GDA User's Guide*
- *GDA Reference Manual*
- *GDA API Reference*

Customer Support Services

You can obtain help with this or any Gensym product from Gensym Customer Support. Help is available online, by telephone, by fax, and by email.

To obtain customer support online:

➔ Access G2 HelpLink at `www.gensym-support.com`.

You will be asked to log in to an existing account or create a new account if necessary. G2 HelpLink allows you to:

- Register your question with Customer Support by creating an Issue.
- Query, link to, and review existing issues.
- Share issues with other users in your group.
- Query for Bugs, Suggestions, and Resolutions.

To obtain customer support by telephone, fax, or email:

➔ Use the following numbers and addresses:

	Americas	Europe, Middle-East, Africa (EMEA)
Phone	(781) 265-7301	+31-71-5682622
Fax	(781) 265-7255	+31-71-5682621
Email	`service@gensym.com`	`service-ema@gensym.com`

1

Using the G2 Dialog Utility

Describes how to use the Gensym Dialog Utility (GDU) module for the rapid building and deployment of Windows dialogs.

Introduction

The G2 Dialog Utility (GDU) module extends the custom standard dialog functionality that G2 provides to enable the rapid building and deployment of native Windows dialogs. GDU provides a mechanism for managing dialogs and dynamically configuring their content, based on a simple grid layout.

When dialogs are displayed at runtime in Telewindows, G2 uses the native controls. Therefore, dialogs specified using GDU can only be displayed within a Telewindows environment. Note that custom dialogs are only supported in

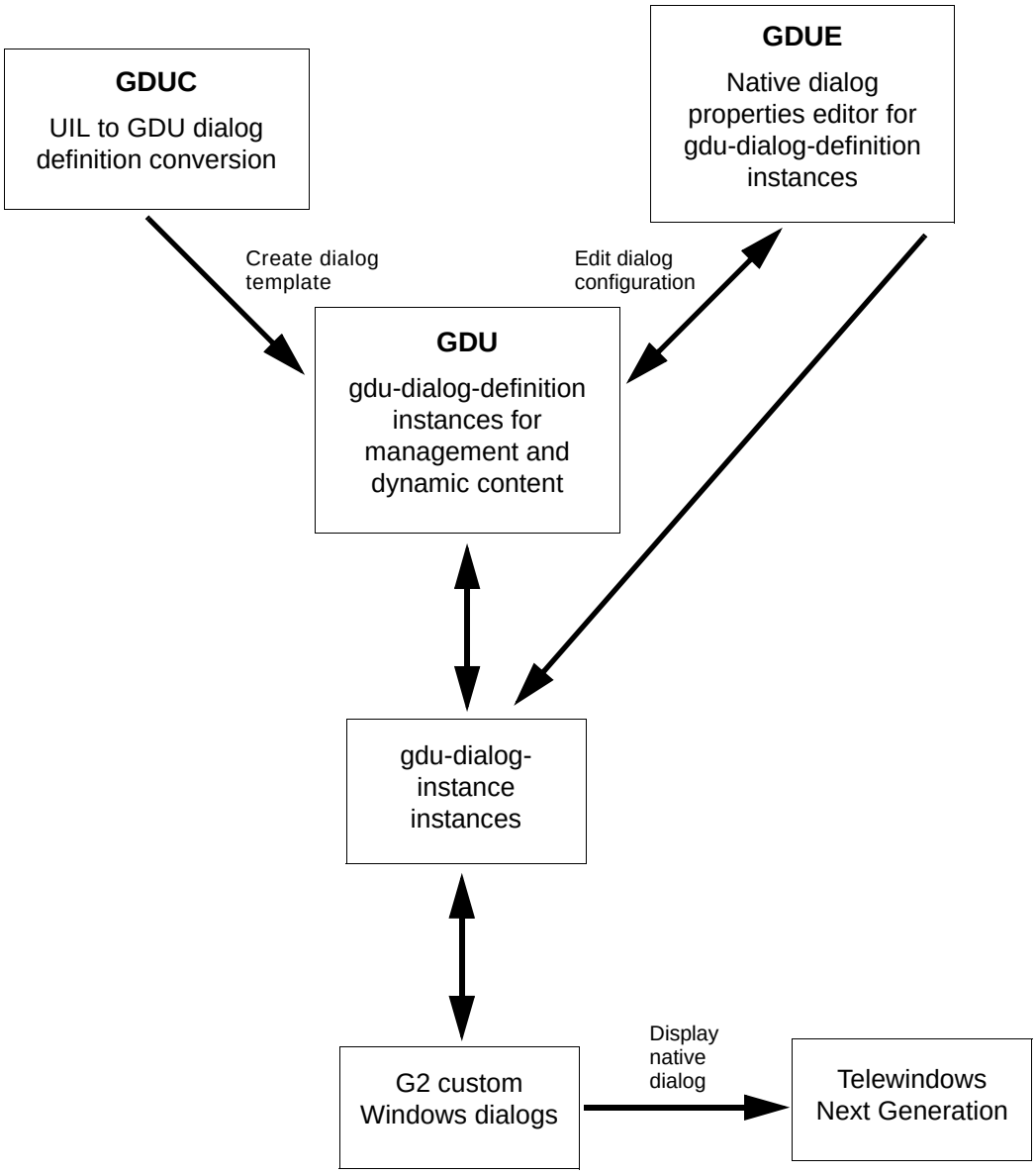
Telewindows Next Generation (`twng.exe`); they are not supported in Telewindows (`tw.exe`).

You can use the GDU module in conjunction with these two other module, which are optional and are typically only included during the development phase of an application:

- G2 Dialog Conversion Utility (GDUC) — Generates gdu-dialog-definition instances, which provide native dialog specifications, based on GUIDE/UIL dialogs. To be fully functional, you need to provide further modifications to those dialogs.
- G2 Dialog Configuration Editor (GDUE) — Configures the properties of a native dialog specification, using the native dialog configuration editor.

Note that the native dialog specifications that the GDUC module generates from UIL dialogs are only templates and required manual editing. For example, you might need to reconfigure the layout or implement logic to manage the dialog, such as enabling or disabling controls based on the context. GDUC does not implement the logic used to manage the content of UIL dialogs.

The following diagram summarizes the relationship of these modules:



For general information on configuring native Windows dialogs, see Chapter 47, "Custom Windows Dialogs" in the *G2 Reference Manual*.

Creating and Configuring Dialog Definitions

The G2 Dialog Utility (GDU) module provides tools for creating and configuring dialog definitions, both manually and dynamically. A dialog definition is an instance of the gdu-dialog-definition class.

Dialogs are typically associated with an object whose properties you want to configure or for which you want to display status information. This object is called the **target object**.

See the `gdu-demo.kb` for examples on how to create and configure dialog definitions.

Creating and Configuring Dialog Definitions Manually

When creating and configuring dialog definitions manually, you only need to configure a subset of the attributes of the dialog definition. You can configure these attributes, using the dialog configuration editor or using the G2 attributes table.

A dialog definition that you configure manually is called a **static dialog**.

By default, the gdu-dialog-instance automatically disables the OK and Apply buttons if a dialog already exists for a particular object and another user opens the same dialog through a different Telewindows client. The dialog remains locked until all dialogs are dismissed.

To create and configure a dialog definition manually:

1 Load or merge in this KB:

Windows	`g2i\kbs\gdu.kb`
UNIX	`g2i/kbs/gdu.kb`

2 Create an instance of a gdu-dialog-definition.

You can create a dialog definition from the GDU palette, which you access by displaying the gdu-top-level workspace and clicking the Palette button:

You can also create a dialog definition by using GDUC. For details, see Chapter 4, “Converting GUIDE Dialogs” on page 109.

3 Configure the following attributes:

Attribute	Description
gdu-dialog-id	A symbolic ID that uniquely identifies the dialog definition.
gdu-dialog-title	The title of the dialog displayed in the caption bar.
gdu-dialog-is-modal	If true, displays the dialog as a modal dialog; otherwise, displays the dialog as a modeless dialog. A modal dialog does not allow editing of other objects until the dialog is closed. The default value is true.
gdu-dialog-is-mdi-child	If true, displays the dialog as an MDI child dialog of the overall window; otherwise, displays the dialog as a modal or modeless dialog. An MDI child dialog has a minimize button and is a modeless dialog. The default value is false.
gdu-dialog-dock	If both gdu-dialog-is-modal and gdu-dialog-is-mdi-child are false and the value of this attribute is not none, the dialog is displayed as a pane. If both gdu-dialog-is-modal and gdu-dialog-is-mdi-child are false and the value of this attribute is none, the dialog is displayed as a modeless dialog. This attribute specifies the docking style of the pane. Valid values are: none, left, right, top, bottom, float and within. The default value is none.
gdu-neighbor-window-name	The name of the neighbor grtl-child-window.
gdu-dialog-neighbor-dock	If the dialog is displayed within a pane and another pane is found on the same side where the dialog should be displayed, this attribute specifies how the dialog should be positioned relative to the neighboring pane. If no neighbor is found, then the dock style specified in gdu-dialog-dock is used. Valid values are: left, right, top, bottom, float and within. Default value is within.
gdu-dialog-is-resizable	If true, dialog is resizable.
gdu-dialog-use-target-icon	If true, the icon of the target object is displayed in the caption bar.

Attribute	Description
gdu-target-class	The class of target objects associated with the dialog.
gdu-dialog-x-position	The x location for the dialog in the window. The default value is center.
gdu-dialog-y-position	The y location for the dialog in the window. The default value is center.
gdu-use-locking	Whether dialog locking is enabled. Default is true.
gdu-lock-procedure	Default is gdu-lock-dialog, which provides the automatic locking behavior described below.
gdu-unlock-procedure	Default is none. To automatically unlock a dialog that is locked once it is dismissed by another user, configure the procedure to be gdu-unlock-dialog.
gdu-allow-multiple-target-dialogs-per-window	When false, the default, brings the existing dialog to the forefront. When true, GDU creates a new dialog. Note that this feature only works if the gdu-dialog-is-mdi-child of the dialog definition is false.

For information about the options for the x and y position of the dialog, see Chapter 47, "Custom Windows Dialogs" in the *G2 Reference Manual*.

4 To configure the contents of the dialog manually, configure these additional attributes:

Attribute	Description
gdu-dialog-width	The width of the dialog. The default value is 245 dialog units.
gdu-dialog-height	The height of the dialog. The default value is 170 dialog units.
gdu-components	The controls to display in the dialog. This attribute contains a sequence of structures, where each structure represents a control. The content of the structure for each control follows the specification of custom Windows dialogs. **Note**: This attribute is hidden in all user modes except administrator mode.

For information on dialog units and the syntax of the components sequence, see Chapter 47, "Custom Windows Dialogs" in the *G2 Reference Manual*.

Creating and Configuring Dialog Definitions Dynamically

To create the content of a dialog dynamically, you specify a component creation procedure in the dialog definition. The signature of this procedure is:

```
my-component-creation-procedure
    (target: class item, dlg: class gdu-dialog-definition,
    win: class g2-window)
```

A dialog definition that you configure by using a component creation procedure is called a **dynamic dialog**.

GDU provides numerous API procedures for dynamically generating the contents and layout of a dialog definition. The component creation procedure should call one or more of the following APIs to add controls to the dialog. Several variations of each API exist for configuring different features of each control.

For the syntax of each of these APIs, see Chapter 3, "Methods for Adding and Manipulating Controls" on page 43.

All the APIs take as standard arguments:

- The dialog in which to display the control.
- The control ID, which is a unique ID for the dialog control. The API formats and displays the control ID as the control prompt label for the control.
- The attribute of the target item whose value the control displays and updates. In general, the attribute of the target item is the initial value for the control.
- Whether the control is initially visible and initially enabled.

Most APIs provide additional versions that allow you to configure whether to add the control in a tab frame, whether to conclude the attribute value immediately, and whether to use a custom validation procedure. For more information, see "Custom Dialog Control Behavior" on page 33.

GDU relies on a grid layout of the controls. The APIs all accept arguments to specify the x-y position within the grid of the control to add, along with its label, if any. It also accepts arguments to specify the number of horizontal and vertical cells that the control and its label should cover within the grid. The APIs generate the contents of the gdu-components attribute and calculate values for the gdu-dialog-width and gdu-dialog-height attributes. For more information, see "Grid Layout Management" on page 20.

When creating and configuring dialogs dynamically, you must configure additional attributes of the dialog component to specify the dialog margins,

spacing, control dimensions, grid dimensions, buttons to include, text resources, and controls to update.

You can use this mechanism to dynamically configure the dialog components and dimensions once, then use the generated definition at runtime. Alternatively, you can use it to dynamically generate the contents of the dialog each time you display it.

See the `gdu-demo.kb` for examples on how to build dialogs dynamically.

To create and configure dialogs dynamically:

1. Create and configure the basic attributes of a gdu-dialog-definition.

   For details, follow steps 1 and 2 under "Creating and Configuring Dialog Definitions Manually" on page 4.

2. Specify a valid G2 procedure or method name in the component-creation-procedure attribute of a gdu-dialog-definition object.

3. Configure the following attributes of the gdu-dialog-definition.

   For a description of the attributes related to the layout of the dialog, see "Grid Layout Management" on page 20.

   These attributes are only relevant if the content of the dialog is generated dynamically, using a component creation procedure:

Attribute	Description
gdu-dialog-caption-height	Specifies the height of the caption of the dialog, which is taken into account when dynamically calculating the height of the dialog. Different themes on your computer may use different heights. The default value is 16.
gdu-dialog-horizontal-margin	The horizontal margin between the first and last columns of controls and the edge of the dialog. Default is 5.
gdu-dialog-vertical-margin	The vertical margin between the first and last rows of controls and the edge of the dialog. Default is 5.
gdu-left-horizontal-spacing	The horizontal spacing between a control and the edge of the left grid cell in which it is located. Default is 5.
gdu-right-horizontal-spacing	The horizontal spacing between a control and the edge of the right grid cell in which it is located. Default is 5.

Attribute	Description
gdu-label-to-control-spacing	The horizontal spacing between the prompt of a control and the control. Default is 2.
gdu-top-vertical-spacing	The vertical spacing between a control and the top edge of the grid cell in which it is located. Default is 4.
gdu-bottom-vertical-spacing	The vertical spacing between a control and the bottom edge of the grid cell in which it is located. Default is 2.
gdu-label-width	The default width used to display prompt labels associated with controls. Default is 50.
gdu-specific-label-width	A sequence that specifies different widths used to display prompt labels for different columns of the grid layout. The first entry specifies the prompt label width for the labels in the first column of the grid layout of controls, the second entry for the second column in the grid, etc. If fewer entries are specified in this list than the x position specified in the gdu-add-* APIs, the default value specified in gdu-specific-label-width is used.
gdu-control-width	The default width used to display controls. Default is 100.
gdu-specific-control-width	A sequence that specifies different widths used to display controls for different columns of the grid layout. The first entry specifies the control width for the controls in the first column of the grid layout of controls, the second entry for the second column in the grid, etc. If fewer entries are specified in this list than the x position specified in the gdu-add-* APIs, the default value specified in gdu-control-width is used.
gdu-grid-row-height	The default height used to display prompt labels associated with controls. Default is 14.
gdu-include-ok-button	If true, adds an OK button to the dialog. Clicking the OK button commits from the controls in the dialog to the attributes of the target object and dismisses the dialog. Default is true.

Attribute	Description
gdu-include-apply-button	If true, adds an Apply button to the dialog. Clicking the Apply button commits from the controls in the dialog to the attributes of the target object. The dialog is not dismissed. Default is true.
gdu-include-update-button	If true, adds an Update button to the dialog. Clicking the Update button refreshes the contents of read-only controls with the current values from the attributes of the target object. Specify the control-id of the controls to refresh in the gdu-control-ids-to-refresh attribute of the dialog specification. To do the refresh, it calls gdu-dialog-refresh-control-values when the user clicks the Update button. Note that if the GRTL module is merged into your application, read-only controls are automatically refreshed periodically so you do not need to add them to the list of controls to refresh. Default is false.
gdu-include-cancel-button	If true, adds a Cancel button to the dialog. Clicking the Cancel button dismisses the dialog without changing the contents of the attributes of the target object. Default is true.
gdu-include-close-button	If true, adds a Close button to the dialog. Clicking the Close button dismisses the dialog without changing the contents of the attributes of the target object. Default is false.
gdu-text-resource	Specifies the GFR resource to use for localizing the dialog. Prompt labels and dialog titles are automatically localized, based on the contents of the specified GFR resource. Default is gdu-text-resources.

Attribute	Description
gdu-control-ids-to-refresh	The sequence of control-id values of the controls to refresh when clicking the Refresh button or when calling gdu-dialog-refresh-control-values. Note that adding a read-only text box, duration control, or timestamp control using the gdu-add-* APIs automatically inserts the control-id of these controls into this list, which you can remove manually, as needed. You can also set the value of this attribute in the component creation procedure.
gdu-dialog-is-closeable	Set to false to disable the close button on the title bar of a dialog. The default value is true. The gdu-show-native-dialog procedure checks the closeable attribute and, if set to false, it calls g2-ui-modify-view to disable the close button.

Displaying, Updating, and Accepting Dialogs

To display the native dialog associated with a dialog definition, you call one of four versions of the gdu-show-native-dialog API procedures. For example, this procedure takes a target item, a dialog definition ID, and a G2 window:

```
gdu-show-native-dialog
    (target: item, id: symbol, win: g2-window )
    -> status: truth-value
```

When you show a native dialog, the dialog definition calls the gdu-get-target-attribute-value method, which gets the current values of the specified attributes of the target object, then updates the dialog components with the new values before displaying the dialog. You can implement custom methods for getting the attribute values of a new target class by implementing a custom gdu-get-target-attribute-value method.

When the user clicks the OK or Apply button, or when a control specifies that the value change should be set immediately, the dialog definition calls the gdu-set-target-attribute-value for each control that needs to set an attribute value on a target object. You can implement custom methods for setting the attribute values of a new target class by implementing a custom gdu-set-target-attribute-value method.

You can update the value of a control on demand or based on a periodic interval. For example, you might need to update the value of a read-only control that displays metrics or status information on a periodic interval. You can update the values of controls in the dialog in one of several ways:

- If the Update button is included in the dialog, the user can refresh the values of controls by clicking this button.
- You can programmatically force an update by calling the gdu-dialog-refresh-control-values method.
- If the GRTL module is merged into your application, periodic updates automatically occur, based on the user-interface-refresh-period setting in the `config.txt` file. For more information, see the *G2 Run-Time Library User's Guide*.

You specify the control ID of the controls to update in the gdu-control-ids-to-refresh attribute of the dialog definition. This attribute specifies a sequence of symbols corresponding to the control IDs to be refreshed. Note that when using the gdu-add-* APIs to add read-only text-box controls, read-only duration controls, or read-only timestamp controls, their control IDs are automatically inserted into the gdu-control-ids-to-refresh sequence.

When the user clicks the OK or Apply button, that is, a control whose control-id attribute is the symbol ok-button or the symbol apply-button, the values of the attributes on the target object are automatically set, based on the values in the controls, except for those marked as read-only. Clicking the OK button also dismisses the dialog.

For the syntax of the other gdu-show-native-dialog methods, see "Show Dialog Methods" on page 35.

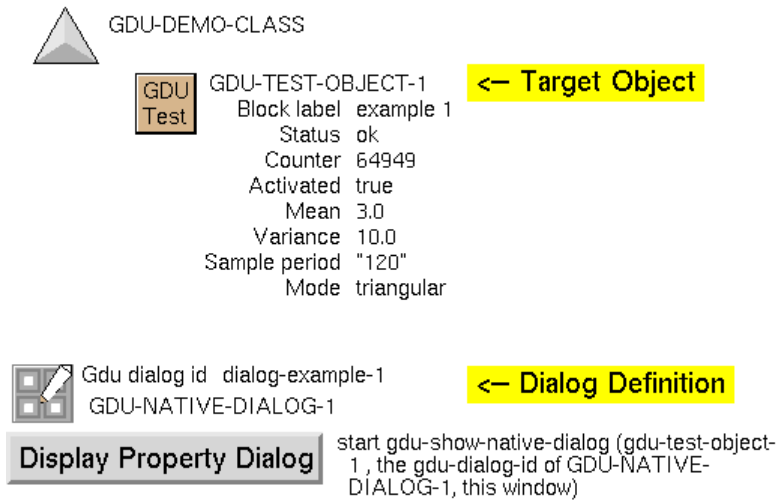
For information in implementing custom get and set methods, see "Custom Get and Set Methods" on page 32.

Example: Displaying a Static Dialog

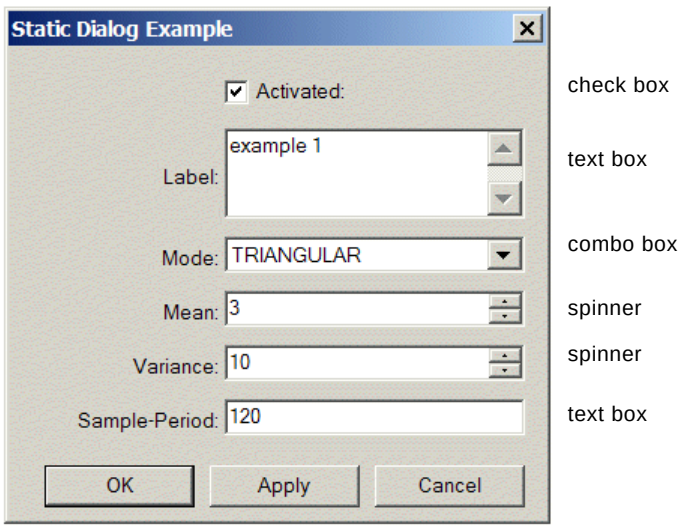
This procedure call shows the native dialog defined by gdu-native-dialog-1 for the target item named gdu-test-object-1 in the current window:

```
start gdu-show-native-dialog
    (gdu-test-object-1, the gdu-dialog-id of gdu-native-dialog-1, this window)
```

Here is the target class, the target item, the dialog definition, and the button that shows the dialog:



Here is the resulting dialog, where the prompt labels appear on the left and the control appears on the right:

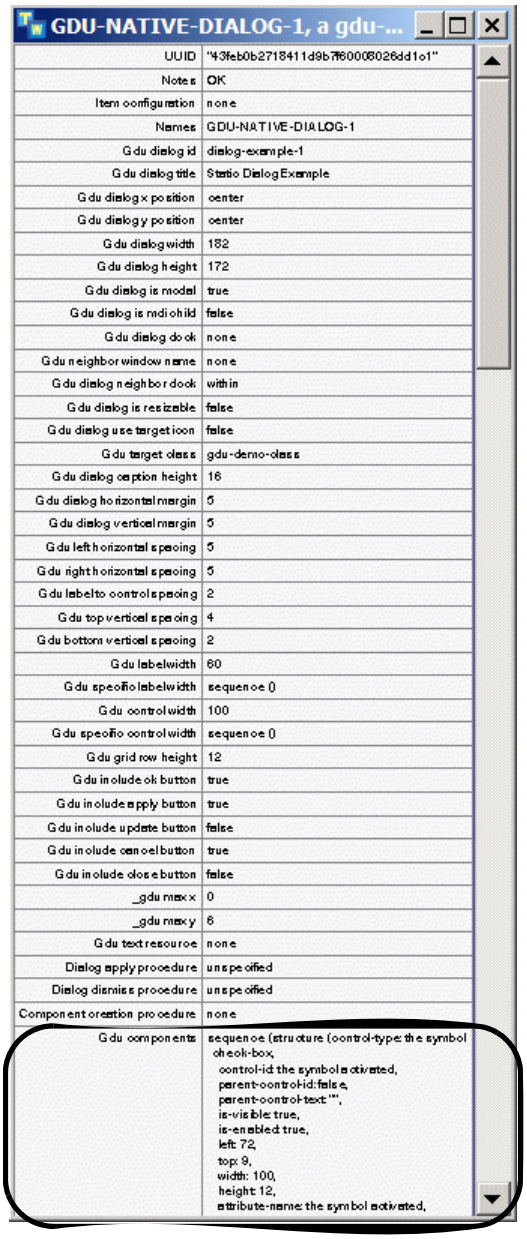


Here is part of the sequence of structures that specifies the dialog components, which were configured manually:

```
sequence
    (structure
        (control-type: the symbol check-box,
        control-id: the symbol activated,
        parent-control-id: false,
        parent-control-text: "",
        is-visible: true,
        is-enabled: true,
        left: 72,
        top: 9,
        width: 100,
        height: 12,
        attribute-name: the symbol activated,
        control-value: structure (text-value: "Activated:"),
        conclude-immediatly: false,
        custom-validate-control-value-procedure: the symbol none,
        response-action: the symbol ignore),
    structure
        (control-type: the symbol label,
        control-id: the symbol label.prompt,
        parent-control-id: false,
        parent-control-text: "",
        is-visible: true,
        left: 10,
        top: 39,
        width: 60,
        height: 12,
        alignment: the symbol right,
        control-value: structure (text-value: "Label:"),
        response-action: the symbol ignore),
    . . . )
```

See Basic Dialog example in `gdu-demo.kb` for the complete example, which is located in the `g2i\examples` directory. On Windows, you can load the demo from the Start menu.

Here is the properties dialog for the gdu-native-dialog-1 dialog definition, which configures the names, dialog ID, title, x and y position, width and height, modal and MDI child attributes, and target class. In this example, the gdu-components attribute is configured manually and, thus, is visible (in administrator mode).

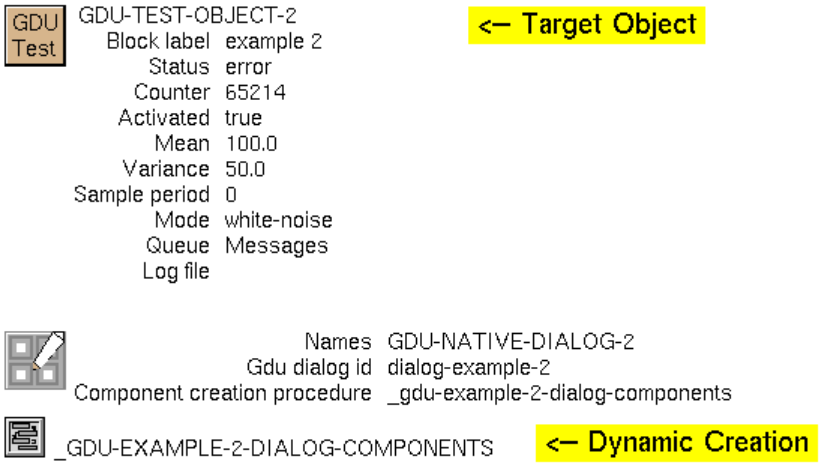


Example: Displaying a Dynamic Dialog

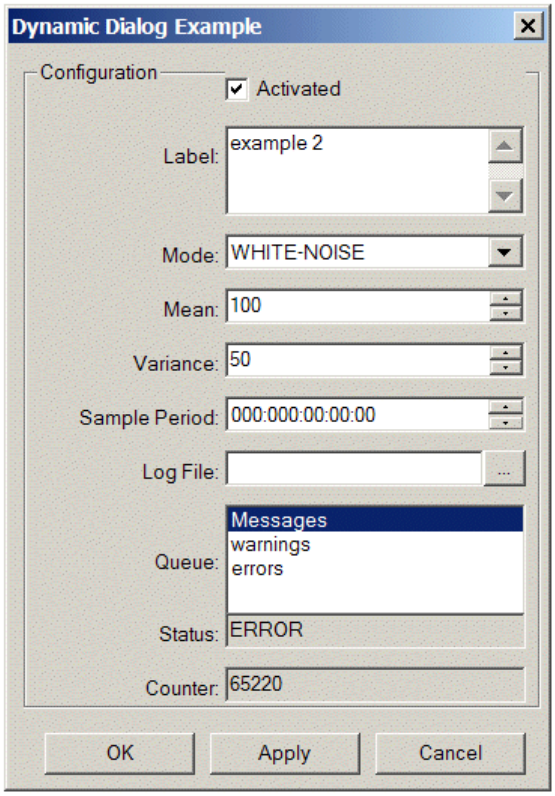
This procedure call shows the native dialog defined by gdu-native-dialog-2 for the target item named gdu-test-object-2 in the current window:

```
start gdu-show-native-dialog
    (gdu-test-object-2, the gdu-dialog-id of gdu-native-dialog-2, this window)
```

Here is the target item, the dialog definition, and the component creation procedure:



Here is the resulting dialog, which includes two read-only controls at the bottom:



group

check box

text box

combo box

spinner

spinner

duration

file selection

list box

read-only text box

read-only text box

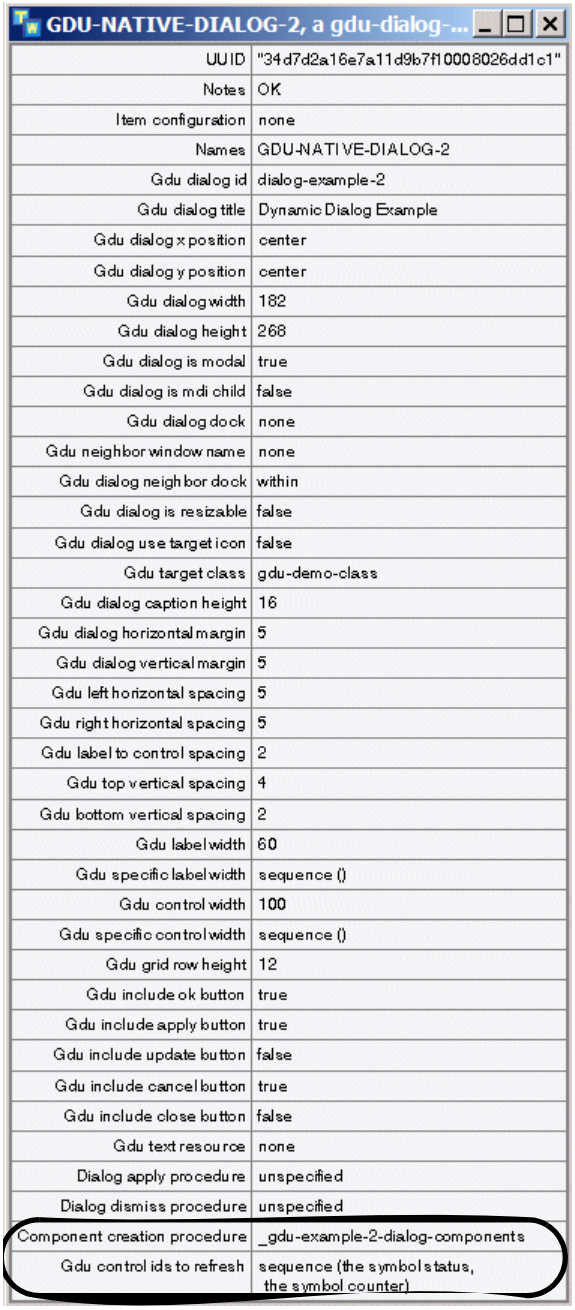
Here is the _gdu-example-2-dialog-components procedure, which dynamically configures the gdu-components of the dialog definition by calling various gdu-add-* API procedures to add dialog components. The procedure also dynamically concludes the gdu-control-ids-to-refresh to allow the specified controls to be updated.

```
_gdu-example-2-dialog-components(Target: class item {target item} ,
    Dlg: class gdu-dialog-definition, Win: class g2-window)

begin

{ --- Build the list of components }
    call gdu-add-check-box-control (Dlg, the symbol activated, the symbol activated,
        true, true, 0, 0, 1, 1);
    call gdu-add-text-box-control(Dlg, the symbol label, the symbol block-label,
        true, true, false, 0, 1, 1, 2);
    call gdu-add-combo-box-control(dlg, the symbol mode, the symbol mode,
        true, true, false, sequence(), the symbol none, 0, 3, 1, 1);
    call gdu-add-spinner-control(Dlg, the symbol mean, the symbol mean,
        true, true, 0.0, 100.0, 1.0, 0, 4, 1, 1);
    call gdu-add-spinner-control(Dlg, the symbol variance, the symbol variance,
        true, true, 0.0, 100.0, 1.0, 0, 5, 1, 1);
    call gdu-add-duration-control (Dlg, the symbol sample-period,
        the symbol sample-period, true, true, 0, 6, 1, 1);
    call gdu-add-file-selection-control (Dlg, the symbol log-file, the symbol log-file,
        true, true, "csv",  sequence(sequence("CSV Files (*.csv)", "*.csv"),
        sequence("All Files (*.*)", "*.*")), true, 0, 7, 1, 1);
    call gdu-add-list-box-control(dlg, the symbol queue, the symbol queue,
        true, true, false, false, sequence( "Messages", "warnings", "errors"),
        the symbol none, 0, 8, 1, 3);
    call gdu-add-text-box-control (Dlg, the symbol status, the symbol status,
        true, true, true, 0, 10, 1, 1);
    call gdu-add-text-box-control (Dlg, the symbol counter, the symbol counter,
        true, true, true, 0, 11, 1, 1);
    call gdu-add-group-control(dlg, the symbol group-1, "Configuration", 0, 0, 1, 12);

{ --- The two metrics values to update dynamically }
    conclude that the gdu-control-ids-to-refresh of dlg =
        sequence(the symbol status, the symbol counter);
end
```

To update the read-only controls specified in the gdu-control-ids-to-refresh, you call the gdu-dialog-refresh-control-values API procedure, as follows:

```
start gdu-dialog-refresh-control-values
    (every gdu-dialog-instance that is a-native-dialog-for-window this window,
    this window)
```

Here is the properties dialog for the gdu-native-dialog-2 dialog definition object, which configures the component creation procedure to be _gdu-example-2-dialog-components. The component creation procedure dynamically configures the gdu-control-ids-to-refresh attribute. The dialog components structure is specified dynamically when the dialog is displayed and, thus, is not shown in the table.



GDU-NATIVE-DIALOG-2, a gdu-dialog-...

UUID	"34d7d2a16e7a11d9b7f10008026dd1c1"
Notes	OK
Item configuration	none
Names	GDU-NATIVE-DIALOG-2
Gdu dialog id	dialog-example-2
Gdu dialog title	Dynamic Dialog Example
Gdu dialog x position	center
Gdu dialog y position	center
Gdu dialog width	182
Gdu dialog height	268
Gdu dialog is modal	true
Gdu dialog is mdi child	false
Gdu dialog dock	none
Gdu neighbor window name	none
Gdu dialog neighbor dock	within
Gdu dialog is resizable	false
Gdu dialog use target icon	false
Gdu target class	gdu-demo-class
Gdu dialog caption height	16
Gdu dialog horizontal margin	5
Gdu dialog vertical margin	5
Gdu left horizontal spacing	5
Gdu right horizontal spacing	5
Gdu label to control spacing	2
Gdu top vertical spacing	4
Gdu bottom vertical spacing	2
Gdu label width	60
Gdu specific label width	sequence ()
Gdu control width	100
Gdu specific control width	sequence ()
Gdu grid row height	12
Gdu include ok button	true
Gdu include apply button	true
Gdu include update button	false
Gdu include cancel button	true
Gdu include close button	false
Gdu text resource	none
Dialog apply procedure	unspecified
Dialog dismiss procedure	unspecified
Component creation procedure	_gdu-example-2-dialog-components
Gdu control ids to refresh	sequence (the symbol status, the symbol counter)

Grid Layout Management

When creating the content of dialogs dynamically, the APIs use a grid layout for managing the location of each control. Each call to gdu-add-* API inserts one or more native controls within the specified cell of the grid layout, depending on the API. For example, the gdu-add-text-box-control API inserts a prompt label and a text box into the specified cell of the grid. A call to gdu-add-file-selection-control inserts a prompt label, a text box, and a push button to launch the file browser into the cell.

You specify the (x, y) position of the cell within the grid in which to place the control, where x = 0, y = 0 is the upper-left cell. For example, to place a control at the top of the second column, you would specify x = 1 (second column) and y = 0 (first row). If the control has a prompt label, the label is on the left and is right aligned, and the control is on the right and is left-aligned within the cell.

You can also specify an x and y span in the gdu-add-* APIs, which enables a component to span more than one cell. Typically, the width of the control extends across more than one cell, not the width of the prompt label. For example, an x span of 2 means the control spans 2 columns in the grid, and a y span of 2 means the control spans 2 rows. Specifying an x and y span of 1 causes the control to span a single column and row.

You specify the overall width of the controls and their associated labels, if any, by configuring these attributes of the dialog definition:

- gdu-specific-label-width
- gdu-control-width
- gdu-specific-control-width
- gdu-dialog-caption-height
- gdu-dialog-horizontal-margin
- gdu-dialog-vertical-margin
- gdu-left-horizontal-spacing
- gdu-right-horizontal-spacing
- gdu-label-to-control-spacing
- gdu-top-vertical-spacing
- gdu-bottom-vertical-spacing
- gdu-label-width
- gdu-specific-label-width
- gdu-control-width

- gdu-specific-control-width
- gdu-grid-row-height

The gdu-specific-label-width and gdu-specific-control-width attributes are sequences of integers specifying the width of the prompt label and control, starting at the first column and for subsequent columns. If fewer values are specified in the sequences than are required for the layout of the dialog, the values specified in gdu-label-width and gdu-control-width are used for the remaining columns.

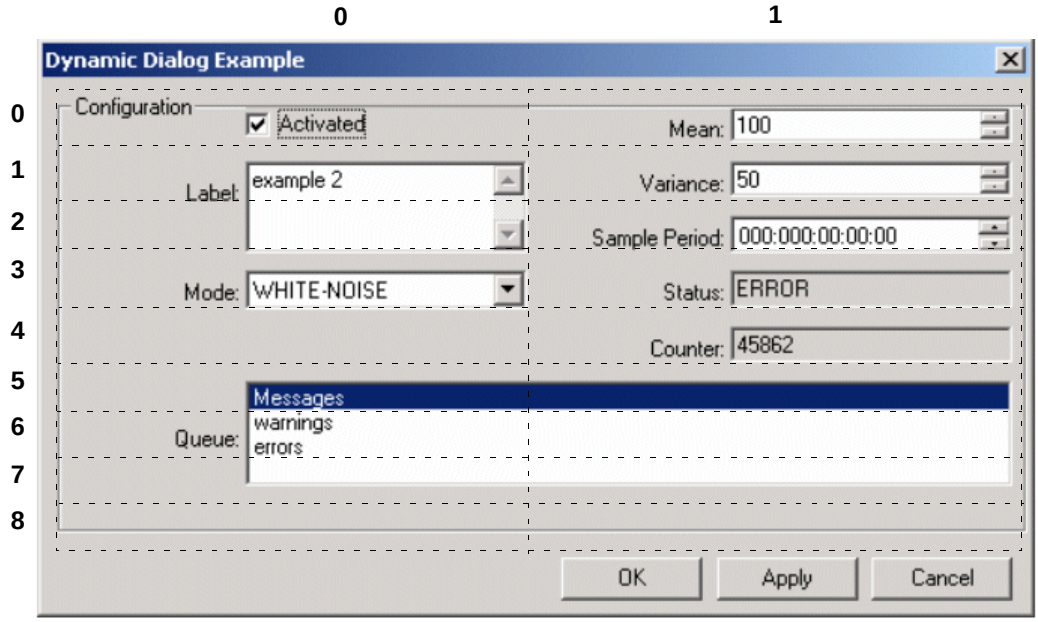
By default, you specify the width and height in dialog units. You can use the following procedures to convert grid coordinates into dialog units:

gdu-label-position-and-dimension
(*dlg*: class gdu-dialog-definition, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)
–> *top*: integer, *left*: integer, *height*: integer, *width*: integer

gdu-control-position-and-dimension
(*dlg*: class gdu-dialog-definition, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)
–> *top*: integer, *left*: integer, *height*: integer, *width*: integer

The dialog width and height is automatically calculated, based on the content, layout, and specifications, as follows.

For example, here is a custom native dialog created by using a component creation procedure, which shows the grid lines used for specifying the layout:



The layout of the dialog is based on a grid of 2 columns and 9 rows:

Control	Grid X, Grid Y	Grid Span X, Grid Span Y
Configuration group	0, 0	2, 8
Activated check box	0, 0	1, 1
Label text box	0, 1	1, 2
Mode combo box	0, 3	1, 2
Mean spinner	1, 0	1, 1
Variance spinner	1, 1	1, 1
Sample period spinner	1, 2	1, 1
Status read-only text box	1, 3	1, 1
Counter read-only text box	1, 4	1, 1
Queue list box	0, 5	2, 3

The component-creation-procedure specifies _gdu-example-2-dialog-components as the procedure to call to dynamically create the dialog:

```
_gdu-example-2-dialog-components(Target: class item {target item},
    Dlg: class gdu-dialog-definition, Win: class g2-window)
begin

{ --- Build the list of components }
    call gdu-add-check-box-control (Dlg, the symbol activated, the symbol activated,
        true, true, 0, 0, 1, 1);
    call gdu-add-text-box-control(Dlg, the symbol label, the symbol block-label, true,
        true, false, 0, 1, 1, 2);
    call gdu-add-combo-box-control(dlg, the symbol mode, the symbol mode, true,
        true, false, sequence(), the symbol none, 0, 3, 1, 1);
    call gdu-add-spinner-control(Dlg, the symbol mean, the symbol mean, true, true,
        0.0, 100.0, 1.0, 1, 0, 1, 1);
    call gdu-add-spinner-control(Dlg, the symbol variance, the symbol variance, true,
        true, 0.0, 100.0, 1.1, 1, 1, 1, 1);
    call gdu-add-duration-control (Dlg, the symbol sample-period, the symbol
        sample-period, true, true, 1, 2, 1, 1);
    call gdu-add-text-box-control (Dlg, the symbol status, the symbol status, true, true,
        true, 1, 3, 1, 1);
    call gdu-add-text-box-control (Dlg, the symbol counter, the symbol counter, true,
        true, true, 1, 4, 1, 1);
    call gdu-add-list-box-control(dlg, the symbol queue, the symbol queue, true, true,
        false, false, sequence("Messages", "warnings", "errors"), the symbol none,
        0, 5, 2, 3);
    call gdu-add-group-control(dlg, the symbol group-1, "Configuration", 0, 0, 2, 8);

'{ --- The two metrics values to update dynamically }
    conclude that the gdu-control-ids-to-refresh of dlg = sequence(the symbol status,
        the symbol counter);
end
```

The following code launches the dialog for a target object named gdu-test-object-2 and displays it on the specified window:

```
start gdu-show-native-dialog (gdu-test-object-2, the symbol dialog-example-2, win)
```

Localizing Dialogs

You can localize the control prompt labels, dialog title, and button text by using a GFR text resource. You use special syntax in the local text resource file to refer to these features of the dialog text.

The gdu-add-* methods also support additional localization, depending on the type of control. For example, you can localize the text for individual radio buttons, and various aspects of the file selection dialog and controls.

To facilitate creating text resource keys for a dialog, you can edit an attribute in the gdu-module-settings object to generate the keys automatically for a dialog.

To localize text that is not otherwise automatically localized, you can call gdu-localize-text.

For general information on localization, see the *G2 Foundation Resources User's Guide*.

See the Localization of Dialogs example in `gdu-demo.kb`, which is located in the `g2i\examples` directory. On Windows, you can load the demo from the Start menu.

Text Resource Keys for Localizing Dialog Features

You can localize the following features of a dialog, using this syntax for the text resource key:

Feature	Text Resource Key	Example
Dialog title	dialog.*gdu-dialog-id*.title	dialog.dialog-example-title
Control prompt	dialog.*gdu-dialog-id.control-id*.prompt	dialog.dialog-example.mode.prompt
Dialog buttons	dialog.*gdu-dialog-id*.ok-button dialog.*gdu-dialog-id*.apply-button dialog.*gdu-dialog-id*.cancel-button dialog.*gdu-dialog-id*.update-button	dialog.dialog-example.ok-button
Check box, push button, and label control text	dialog.*gdu-dialog-id.control-id*	dialog.dialog-example.activated
Radio button choices	dialog.*gdu-dialog-id.control-id-n* where *n* is a 0-based index of the radio buttons as specified in the control, where 0 is the first radio button specified, 1 is the second, and so on.	dialog.dialog-example.my-radio-buttons-0 dialog.dialog-example.my-radio-buttons-1 dialog.dialog-example.rmy-adio-buttons-2

Feature	Text Resource Key	Example
Combo box and list box choices	dialog.*gdu-dialog-id*.choice.*symbol* where *symbol* is the text choice converted to a symbol by replacing spaces with hyphens. For example, if the *list-of-choices* is sequence("choice 1", "choice 2"), the symbols would be choice-1 and choice-2. To localize the list of choices, set the *localize-list-of-choices* argument to the gdu-add-* APIs to true.	dialog.dialog-example.choice.choice-1 dialog.dialog-example.choice.choice-2
File selection control	Button text for displaying file selection dialog (default: ...) dialog.*dialog-id*.*control-id*.browse	dialog.dialog-example.file.browse
	File selection dialog title text (default: Select File) dialog.*dialog-id*.*control-id*.title	dialog.dialog-example.file.title
	File selection dialog OK button text (default: Select) dialog.*dialog-id*.*control-id*.ok	dialog.dialog-example.file.ok
	File selection dialog Cancel button text (default: Cancel) dialog.*dialog-id*.*control-id*.cancel	dialog.dialog-example.file.cancel
	Confirmation dialog title text (default: File Selection) dialog.*dialog-id*.*control-id*.confirm.caption	dialog.dialog-example.file.confirm.caption
	Confirmation dialog label text (default: File already exists. Do you want overwrite it?) dialog.*dialog-id*.*control-id*.confirm.label	dialog.dialog-example.file.confirm.label

Localizing Dialog Text

To localize dialog text:

1. Create a GFR text resource and a local text resource.

   For details, see the *G2 Foundation Resources User's Guide*.

2. Configure the gdu-text-resource attribute of the dialog definition to refer to the GFR text resource to use for localizing dialog text.

3. In the local text resource file of the local text resource, use the special syntax described in "Text Resource Keys for Localizing Dialog Features" on page 24 to localize the various features of the dialog.

   You can also automatically generate text resource keys for all features of the dialog that can be localized. For details, see "Automatically Generating Text Resource Keys for a Dialog" on page 30.

Tip To configure the local text resource, merge `gxl.kb` into your application, and enable the Array and List Editing option on the gxl-top-level workspace, then choose edit resources on the local text resource. Alternatively, you can configure a text file and import it into the text localization object. For details, see the *G2 Foundation Resources User's Guide*.

---

4. Configure the gdu-components attribute of the dialog definition to use the lookup key.

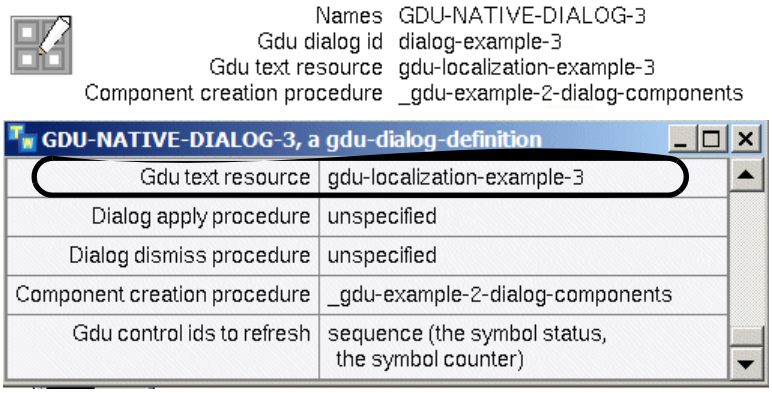
   For example:

   ```
   control-id: the symbol label.prompt
   ```

   Alternatively, you can use a component-creation-procedure and call one of the gdu-add-* APIs, which automatically uses this syntax to refer to the key for looking up text values for prompt labels in the local text resource.

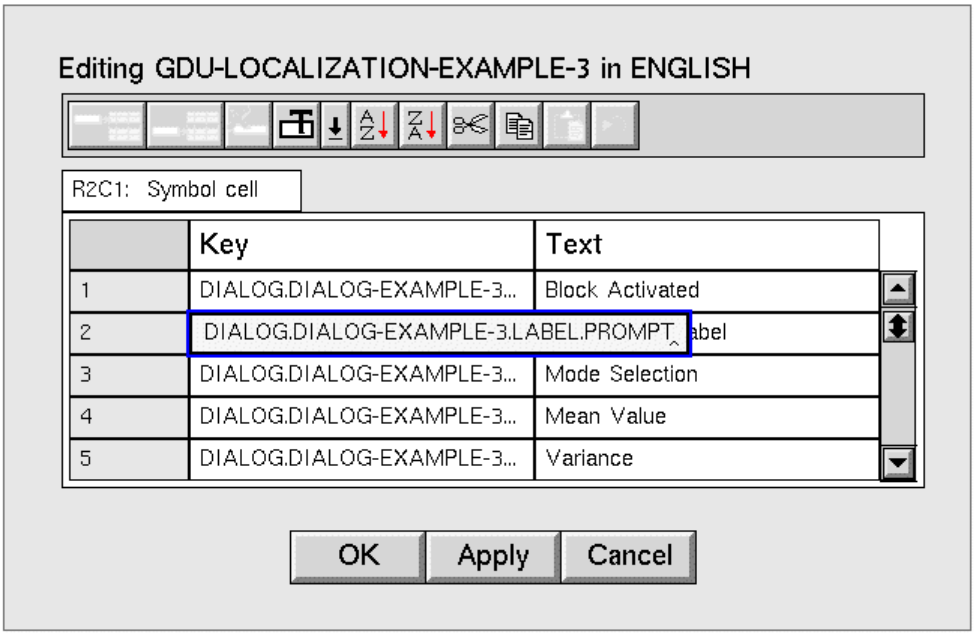
Example: Localizing Text Prompts

This example shows how to local text prompts in a dialog that uses a component creation procedure to create dialog components dynamically. Here is the gdu-dialog-definition, which specifies its gdu-text-resource as gdu-localization-example-3:

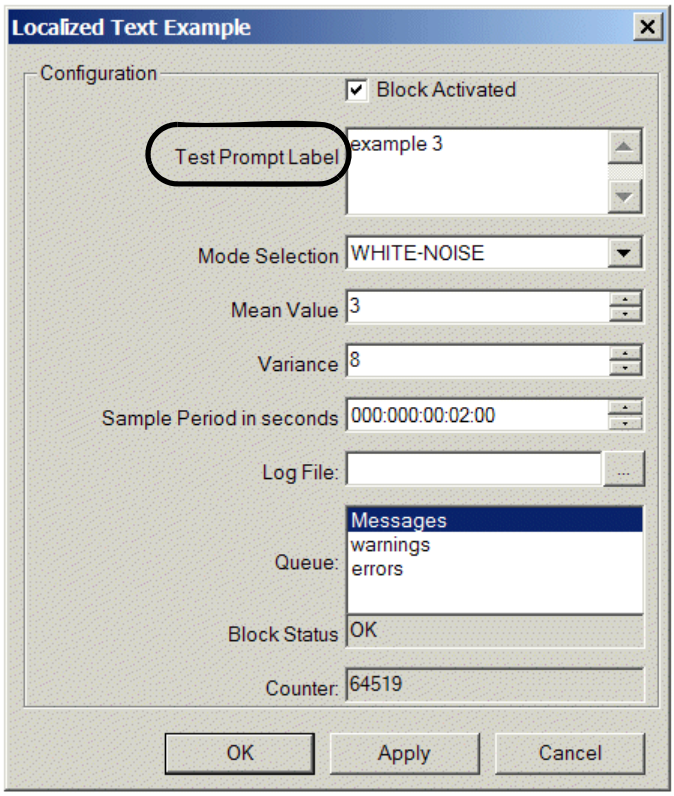


Here is the GFR text resource named gdu-localization-example-3 and its associated local text resource:

Here is the spreadsheet for editing the local text resource, which shows the keys and text values for each control whose prompts should be localized. For example, the text value for the dialog.dialog-example-3-label.prompt is Test Prompt Label.



Here is resulting dialog with the localized text:



Here is the portion of the gdu-components specification in the dialog definition, which is automatically configured to refer to the label.prompt as the control-id of the label control:

```
structure (control-type: the symbol label,
   control-id: the symbol label.prompt,
   parent-control-id: false,
   parent-control-text: "",
   is-visible: true,
   left: 10,
   top: 33,
   width: 100,
   height: 24,
   alignment: the symbol right,
   control-value: structure (text-value: "Label:"),
   response-action: the symbol ignore)
```

Automatically Generating Text Resource Keys for a Dialog

Rather than having to create text resource keys for those features of a dialog that you want to localize, you can automatically generate text resource keys for all features of the dialog, which you can then edit, as needed.

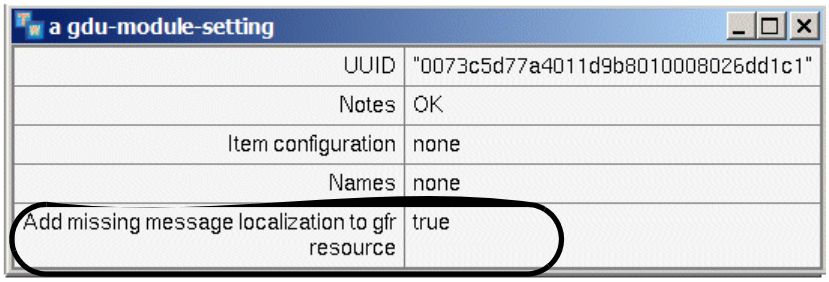
You do this by creating a GFR text resource and associated local text resource for a dialog, and editing an attribute in the gdu-module-settings object. The first time you display the dialog, GDU automatically generates text resource keys in the local text resource for all features of the dialog that you can localize.

To automatically generate text resource keys for a dialog:

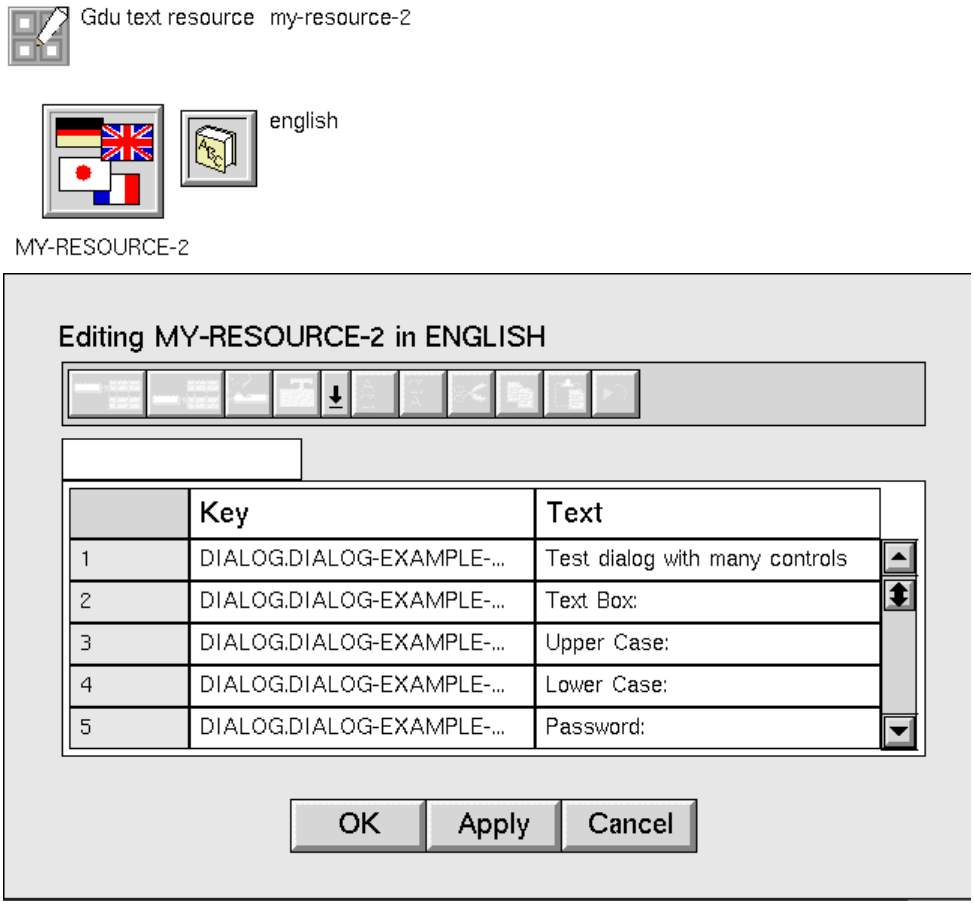
1. Go to the gdu-top-level workspace.
2. Click the Settings button to display the gdu-module-settings dialog for the GDU module.
3. Clone the module settings object and place it on a workspace in a higher-level module to make permanent changes, or edit the GDU module settings object to make temporary changes.
4. Set the add-missing-message-localization-to-gfr-resource attribute to true.
5. Create a GFR text resource and a local text resource.
6. Configure the gdu-text-resource attribute of the dialog definition to refer to the GFR text resource to use for localizing dialog text.
7. Display the dialog whose features you want to localize.

The local text resource is automatically configured with all possible text resource keys for the dialog.

Here is the gdu-module-settings object and its table:



Here is the gdu-dialog-definition object, which specifies gdu-text-resource as my-resource-2, the GFR text resource, and its local text resource. The keys in the table for the local text resource are automatically generated when you display the dialog.



Calling an API Procedure to Localize Text Explicitly

If you need to explicitly localize text in a custom procedure, call this API:

gdu-localize-text
(*dlg*: gdu-dialog-definition, *key*: symbol, *default-text*: text,
win: g2-window)
–> *localized-text*: text

Localizes the text specified by *key*. The procedure first attempts to get a localized text string from the GFR resource specified in the gdu-text-resource of the dialog definition, using the current language of the g2-window. If no value is found, the procedure uses the *default-text* argument; otherwise, the

procedure converts the *key* to a text string and returns it as the localized text. Use the following syntax as the key in the local text resource:

dialog.*gdu-dialog-id.key*

For example:

dialog.dialog-example-1.label

Customizations

You can specify the following customizations when configuring the dialog components dynamically:

- Custom get and set methods.
- Custom apply and dismiss behavior.
- Custom dialog control behavior.

For examples of customizations, see the `gdu-demo.kb`, which is located in the `g2i\examples` directory. On Windows, you can load the demo from the Start menu.

Custom Get and Set Methods

The most common customization is to write custom methods that get and set the attribute of a target object. The default get and set methods are called each time a control in the dialog needs to get a value from the attribute to update the value in the control and when the control needs to set the value of the control for the attribute of the target object.

Using custom get and set methods provides great flexibility, because it means you can use "virtual attributes" as the attribute of the target object to display and update in the dialog. For example you can use custom get and set methods to perform data transformations, whereby the control might supply a value as a text, but you might want to store the value as a symbol or quantity.

GDU provides generic implementations of the following get and set methods for the item class:

gdu-get-target-attribute-value
(*target*: class item, *attribute-name*: symbol, *dlg*: class gdu-dialog-definition, *win*: class g2-window)
–> *val*: value

gdu-set-target-attribute-value
(*target*: class item, *attribute-name*: symbol, *val*: value, *dlg*: class gdu-dialog-definition, *win*: class g2-window)

For an example, see the Custom Attribute Getter/Setter example in `gdu-demo.kb`.

Custom Apply and Dismiss Behavior

The dialog definition object allows you to specify custom procedures that the dialog calls when the user accepts and dismisses the dialog. You specify these procedures in the following attributes of the dialog definition object:

- dialog-apply-procedure — Called when the user clicks the Apply button in the dialog. The signature of this procedure is:

  my-apply-procedure
  (*dlg*: class gdu-dialog-instance, *target*: class item,
  control-values: sequence, *win*: class g2-window)

- dialog-dismiss-procedure — Called when the user clicks the OK button in the dialog. If the user clicks the OK button, which also accepts the dialog, this procedure is called second. The signature of this procedure is:

  my-dismiss-procedure
  (*dlg*: class gdu-dialog-instance, *target*: class item,
  dialog-was-accepted: truth-value, *control-values*: sequence,
  win: class g2-window).

Custom Dialog Control Behavior

The structure for each control in the gdu-components of a gdu-dialog-definition object can specify additional attributes to customize its behavior and manage dialogs. You can configure the following attributes by using the dialog configuration editor.

- custom-configuration-procedure — The symbolic name of a procedure that is called when the dialog is configured but before it is displayed. You use this procedure to configure each control, such as localization of labels or the initial value of the control, based on the attribute of the target object. The signature of the procedure is:

  control-configuration-procedure
  (*dlg*: class gdu-dialog-definition, *control*: structure, *target*: class item,
  win: class g2-window)
  –> *control-structure*: structure

- custom-set-control-value-to-target-procedure — The symbolic name of a procedure that is called when the values of a control need to be concluded to the attribute values of the target object. The signature of this procedure is:

  set-control-value-procedure
  (*dlg*: class gdu-dialog-instance, *control*: structure,
  new-value: structure, *target*: class item, *win*: class g2-window)

  We recommend creating custom get and set methods on the target object, rather than specifying a custom-configuration-procedure and custom-set-

control-value-to-target-procedure. For details, see "Custom Get and Set Methods" on page 32.

- custom-update-control-value-from-target-procedure — The symbolic name of a procedure that is called when the values of controls need to be updated with the values from the target object, such as when updating metrics values or status information. The signature of this procedure is:

  set-control-value-procedure
  (*dlg*: class gdu-dialog-instance, control: structure, *target*: class item, *win*: class g2-window)
  –> *control-action-structure*: structure

  This procedure should return an empty structure or a control action structure applicable for the control, for example, actions such as add or replace. For more information, see the description of modifying a control in Chapter 47, "Custom Windows Dialogs" in the *G2 Reference Manual*.

The structure for each control in the gdu-components of a gdu-dialog-definition object can also specify the following attributes. These attributes are automatically set when using the gdu-add-* APIs to add controls to a dialog.

- attribute-name — The attribute name of the G2 object, as a symbol. In the future, this attribute will support the dot notation to refer to attributes on subobjects.
- conclude-immediately — If true, the value is set immediately when changed. The response-action on the control also needs to be set correctly so that updates are sent to the server as they change, for example, using respond or respond-with-all-data. That way, changes always go through the validation process when the value is committed to the attribute of the target object. The default value in the control structure is false.

  For information on configuring the response-action, see Chapter 47, "Custom Windows Dialogs" in the *G2 Reference Manual*.
- on-value — For radio-button controls, if the current value of the attribute of the target object is equal to the on-value, the radio button is checked, otherwise it is unchecked. The default value in the control structure is none.
- choices-procedure — For combo-box and list-box controls, the procedure name to call to get the list of choices. The default value in the control structure is none. The signature of this procedure is:

  choices-procedure
  (*target*: class item, *attribute-name*: symbol, *win*: class g2-window)
  –> *choices*: sequence

- custom-control-activation-procedure — For push-button controls, the procedure name that is called when the button is pressed. The default value in the control structure is none. The signature of this procedure is:

  activation-procedure
  (*dlg*: class gdu-dialog-instance, *control*: structure,
  other-values: sequence, *target*: class item, *win*: class g2-window)"

- custom-validate-control-value-procedure — The procedure that is called when the value changes and the response-action notifies the server to validate the value. Note that if this procedure is specified, the response-action of the control is automatically set to respond-with-all-data. The default value in the control structure is none. The signature of this procedure is:

  validation-procedure
  (*dlg*: class gdu-dialog-instance, *control*: structure,
  other-values: sequence, *target*: class item, *win*: class g2-window)
  –> *valid*: truth-value

  Other values refer to the values of other controls, and the control-value of the control is updated with the current value received from the client control before the validation procedure is called. This procedure should return true if the value is valid, false otherwise. Note that this procedure can also enable, disable, show, or hide other controls by calling the gdu-dialog-enable-control, gdu-dialog-disable-control, gdu-dialog-show-control, and gdu-dialog-hide-control APIs. For example, when a check box is unchecked, you can disable attributes that are no longer relevant.

For examples, see the `gdu-demo.kb`.

Show Dialog Methods

The following methods are defined on the item class and allow you to show a native dialog for a target item. Each version of the show-native-dialog method shows a native dialog for a target item on a given window. The methods differ in the additional arguments that you can provide to specify which dialog definition object to use.

For each method, prior to displaying the dialog, each component is updated to localize text labels and set the initial value to the value of the object. You can specify a custom control update procedure, as a symbol, in the custom-configuration-procedure attribute of the gdu-components attribute structure of each control in the dialog definition object. For a description of the custom configuration procedure, see “Custom Dialog Control Behavior” on page 33.

If the component-creation-procedure attribute of the dialog definition refers to a valid procedure or method name, prior to displaying the dialog, the component creation procedure is called to dynamically determine the list of components. Otherwise, if no component creation procedure is specified, the dialog uses the

static definition specified in the gdu-components attribute of the dialog definition. For a description of the component creation procedure, see "Creating and Configuring Dialog Definitions Dynamically" on page 7.

When the user clicks the Apply button, the show-native-dialog method:

1. Sets the values on the attributes of the target object.
2. Calls the procedure specified in the dialog-apply-procedure attribute of the dialog definition.

   For the syntax of this method, see "Custom Apply and Dismiss Behavior" on page 33.
3. Updates the content of read-only controls by calling gdu-dialog-refresh-control-values.

When the user clicks the OK button, the show-native-dialog method:

1. Dismisses the dialog.
2. Sets the values on the attributes of the target object, if dialog-was-accepted is true. The dialog-was-accepted attribute is false if the dialog was cancelled, otherwise, it is true.
3. Calls the procedure specified in the dialog-dismiss-procedure attribute of the dialog definition.

   For the syntax of this method, see "Custom Apply and Dismiss Behavior" on page 33.

Here are the four versions of the show-native-dialog method:

gdu-show-native-dialog
(*target*: item, *win*: g2-window)
–> truth-value

Shows a native dialog for a target item, which is an instance of gdu-dialog-definition whose gdu-target-class is equal to the target item's class.

gdu-show-native-dialog
(*target*: item, *id*: symbol, *win*: g2-window)
–> truth-value

Shows a native dialog for a target item, given the *id*, which is the gdu-dialog-id of a gdu-dialog-definition.

gdu-show-native-dialog
(*target*: item, *id*: symbol, *dialog-instance-class-name*: symbol, *win*: g2-window)
–> truth-value

Shows a native dialog for a target item, given the *id* and *dialog-instance-class-name*. The *dialog-instance-class-name* is a symbol naming the dialog instance class name, typically, gdu-dialog-instance, or a subclass.

gdu-show-native-dialog
(*target*: item, *dlg-definition*: gdu-dialog-definition,
dialog-instance-class-name: symbol, *wait-until-dialog-dismissed*: truth-value,
win: g2-window)
–> truth-value

Shows a native dialog for a target item for a given *dlg-definition* and *dialog-instance-class-name*. If *wait-until-dialog-dismissed* is true, the method waits until the dialog is dismissed and returns true if the user clicks the OK button and false if the user clicks the Cancel button. If wait-until-dialog-dismissed is false, the method immediately returns false.

For examples, see the `gdu-demo.kb`, which is located in the `g2i\examples` directory. On Windows, you can load the demo from the Start menu.

Dialog Instance Methods

The following methods are defined on the gdu-dialog-instance class.

gdu-dialog-get-handle
(*dlg*: gdu-dialog-instance, *win*: g2-window)
–> *handle*: integer

Returns the integer handle of the specified gdu-dialog-instance.

gdu-dialog-control-value-changed
(*dlg*: gdu-dialog-instance, *target*: item, *reason*: symbol,
control-id: symbol, *new-control-value*: value, *other-values*: sequence,
win: ui-client-item)

Called when a value has changed in the dialog and the response-action for the specified control is set to notify the server by using respond or respond-with-all-data. *Control-id* is the control that triggers the change. For information on configuring the response-action, see Chapter 47, "Custom Windows Dialogs" in the *G2 Reference Manual*.

gdu-dialog-dismissed
(*dlg*: gdu-dialog-instance, *target*: item, *dialog-was-accepted*: truth-value,
new-values: sequence, *win*: ui-client-item)

Called when the dialog is dismissed, such as when the user clicks the OK or Cancel button. This method sets the dialog-was-accepted attribute to true if the user clicks OK and false if the user clicks Cancel.

gdu-dialog-refresh-control-values
(*dlg*: gdu-dialog-instance, *win*: ui-client-item)

Called to auto refresh the controls listed in the gdu-control-ids-to-refresh attribute of the dialog. The value of the control is updated with the value of

the attribute of the target. Only the specified controls are updated, which avoids unnecessary overhead and updates.

gdu-dialog-disable-control
(*dlg*: gdu-dialog-instance, *control-id*: value)

Disables the specified control. Note that the change is cached first, then all changes are sent at once.

gdu-dialog-enable-control
(*dlg*: gdu-dialog-instance, *control-id*: value)

Enables the specified control. Note that the change is cached first, then all changes are sent at once.

gdu-dialog-hide-control
(*dlg*: gdu-dialog-instance, *control-id*: value)

Hides the specified control. Note that the change is cached first, then all changes are sent at once.

gdu-dialog-show-control
(*dlg*: gdu-dialog-instance, *control-id*: value)

Shows the specified control. Note that the change is cached first, then all changes are sent at once.

gdu-dialog-send-queued-update-actions
(*dlg*: gdu-dialog-instance, *win*: g2-window)

Sends any pending dialog update requests. If this method is not called, an automatic update is performed after the dialog has been displayed and after processing any dialog modification requests.

The APIs to show, hide, enable, disable, and update controls queue up the requests but do not send them at that the time. The reason is that multiple changes to the dialog might be requested at once, and the requests are performed together for optimal efficiency and visual impact.

In addition, this method is defined on a gdu-dialog-definition given a control ID to return the control structure of the dialog instance:

gdu-dialog-get-control
(*dlg*: class gdu-dialog-definition, *control-id*: symbol)
–> *control-structure*: structure

Returns the control structure of the specified gdu-dialog-definition given the *control-id*, or an empty structure if none exists.

Get and Set Methods

The following methods are defined on the class item and provide default implementations for getting and setting an attribute value of a target item for a given dialog definition.

For information on customizing these methods, see "Custom Get and Set Methods" on page 32.

gdu-get-target-attribute-value
(*target*: item, *attribute-name*: symbol, *dlg*: gdu-dialog-definition, *win*: g2-window)
–> *val*: value

Returns the current value of an attribute of a target item for a given dialog definition.

Note that if GRTL is merged into your application, this method calls grtl-get-property-value to get the current value. Therefore, if GRTL is merged in, we recommend that you write custom grtl-get-property-value methods to ensure consistency since other modules use the get and set methods.

gdu-set-target-attribute-value
(*target*: item, *attribute-name*: symbol, *val*: value, *dlg*: gdu-dialog-definition, *win*: g2-window)

Sets the value of an attribute of a target item to a new value for a given dialog definition.

Note that if GRTL is merged into your application, this method calls grtl-set-property-value to get the current value. Therefore, if GRTL is merged in, we recommend that you write custom grtl-set-property-value methods to ensure consistency since other modules use the get and set methods.

2

Module Settings

Describes the Gensym Dialog Utility (GDU) module settings.

Introduction

The gdu-module-settings object inherits GFR module settings. Upon startup, GFR locates one module settings object as the active setting, which is typically the instance in the highest level module. The active module is determined when G2 is started. Several APIs take the active module settings object into account during execution.

gdu-module-settings

Manages system configurations for the GDU module.

Class Inheritance Path

gfr-module-settings, object, item

Attributes

Attribute	Description
add-missing-message-localization-to-gfr-resource	Whether to automatically generate text resource keys in the local text resource for all features of the dialog that you can localize.
Allowable values:	truth-value
Default value:	false
Notes:	See "Automatically Generating Text Resource Keys for a Dialog" on page 30.

3

Methods for Adding and Manipulating Controls

Describes the GDU methods for adding and manipulating specific dialog controls.

Introduction

The following methods are defined on the gdu-dialog-definition class. These methods allow you to dynamically add controls to the gdu-components attribute of the dialog definition.

The methods add a control to a given dialog. You specify a unique control ID, which the dialog formats as its prompt label. You also specify the attribute name of the target item that is the attribute whose value the control displays and updates. The attribute of the target item is also the initial value for the control.

The method takes the following standard arguments:

Argument	Description
dlg	The gdu-dialog-definition to which to add the control.
control-id	A unique ID for the control, as a symbol. Within a single dialog, each control must have a unique ID. For those controls that include prompt label, the API formats the control ID by replacing hyphens with spaces and capitalizing each word.
attribute-name	The attribute name of the target item whose value the control displays and updates, as a symbol. The value of the specified attribute name is typically the initial value for the control. See attribute-name in "Custom Dialog Control Behavior" on page 33.
initially-visible	If true, display the control when the dialog is initially displayed.
initially-enabled	If true, enable the control when the dialog is initially displayed.
conclude-immediately	If true, set the value immediately when changed. See conclude-immediately in "Custom Dialog Control Behavior" on page 33.
validation-procedure	The procedure that is called when the value changes and the response-action is configured to notify the server to validate the value. See custom-validate-control-value-procedure in "Custom Dialog Control Behavior" on page 33. For information on configuring the response-action, see Chapter 47, "Custom Windows Dialogs" in the *G2 Reference Manual*.

Argument	Description
anchor	Used in resizable dialogs. The options are one of the symbols none, top, left, bottom, right, top-left, bottom-right, top-left-bottom-right, a sequence of these symbols, or an integer bitmask. The default value is top-left. For a description of how the anchor options behave, see the Microsoft .NET documentation at: `http://msdn2.microsoft.com/en-us/library/system.windows.forms.anchorstyles(vs.71).aspx`
grid-x	The x value that describes the location of the control in the dialog grid. See "Grid Layout Management" on page 20.
grid-y	The y value that describes the location of the control in the dialog grid. See "Grid Layout Management" on page 20.
grid-x-span	The number of cells in the x dimension that the controls spans. See "Grid Layout Management" on page 20.
grid-y-span	The number of cells in the y dimension that the control spans. See "Grid Layout Management" on page 20.
tab-control	The tab control ID of the tab frame in which to display the control, if any, as a value. See "Tab Frame" on page 87.
tab-name	The tab name of the tab frame in which to display the control, as a string. See "Tab Frame" on page 87.

For additional information on using these methods, see:

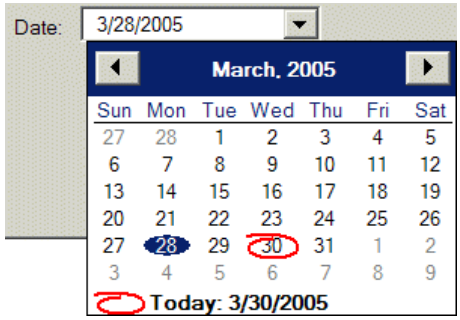
- "Creating and Configuring Dialog Definitions Dynamically" on page 7.
- "Example: Displaying a Dynamic Dialog" on page 16.
- Chapter 47, "Custom Windows Dialogs" in the *G2 Reference Manual*.

For examples, see the `gdu-demo.kb`, which is located in the `g2i\examples` directory. On Windows, you can load the demo from the Start menu.

Calendar

gdu-add-calendar-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *long-date-format*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a calendar control to the dialog, including a prompt label. Set *long-date-format* to true to display the date in a long format, such as Friday, April 14, 2005, or false, to display it in a short format, such as 4/19/05. For example:



See the Many Controls example in `gdu-demo.kb`.

gdu-add-calendar-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *long-date-format*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a calendar control to the dialog in a tab frame.

gdu-add-calendar-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *long-date-format*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a calendar control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-calendar-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *long-date-format*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a calendar control to the dialog with an anchor position when used in resizable dialogs. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

Check Box

gdu-add-check-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a check box control to the dialog, including a prompt label. The check box appears to the left and the label to the right. For example:

You can provide localized text for the prompt label by referring directly to the control ID as the key. For details, see "Localizing Dialogs" on page 23.

See the Dynamic Dialog Specification example in `gdu-demo.kb`.

gdu-add-check-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a check box control to the dialog in a tab frame.

gdu-add-check-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a check box control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-check-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Set *use-cell-width* to true to use the full width of the grid cell for the control, which left-aligns the control in the cell. Specifies *conclude-immediately* and *validation-procedure*.

gdu-add-check-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a check box control to the dialog in a tab frame, and specifies *use-cell-width*, *conclude-immediately*, and *validation-procedure*.

gdu-add-check-box-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a check-box control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *use-cell-width*, *conclude-immediately*, and *validation-procedure*.

gdu-add-multiple-check-box-controls
(*dlg*: gdu-dialog-definition, *control-ids*: sequence, *attribute-names*: sequence, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a sequence of check box controls to the dialog, including prompt labels. For example:

See the Many Controls example in `gdu-demo.kb`.

gdu-add-multiple-check-box-controls
(*dlg*: gdu-dialog-definition, *control-ids*: sequence, *attribute-names*: sequence, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a sequence of controls to the dialog in a tab frame.

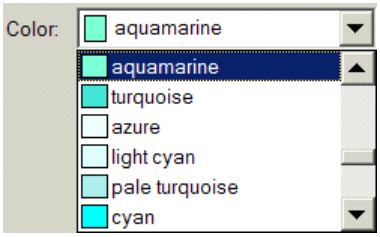
gdu-add-multiple-check-box-controls
(*dlg*: class gdu-dialog-definition, *control-ids*: sequence, *attribute-names*: sequence, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *anchor*: value, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a sequence of check-box controls to the dialog with an anchor position when used in a resizable dialog. Also adds the controls in a tab frame, and specifies *use-cell-width*, *conclude-immediately*, and *validation-procedure*.

Color Picker

gdu-add-color-selection-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a color picker control to the dialog, including a prompt label. The list of colors is determined by the value of the colors-on-2nd-level-color-menu in the Color Parameters system table, whose options are all, standard set, or a list of colors. This attribute determines the colors on the second-level color menu, which you access by choosing More on a G2 color palette. For example:



See the Many Controls example in `gdu-demo.kb`.

gdu-add-color-selection-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a color picker control to the dialog in a tab frame.

gdu-add-color-selection-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a color picker control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-color-selection-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *list-height*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Specifies *list-height*, *conclude-immediately*, and *validation-procedure*. Set *list-height* to the number of rows to display in the control. The default value is 6.

gdu-add-color-selection-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *list-height*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a color picker control to the dialog in a tab frame, and specifies *list-height*, *conclude-immediately*, and *validation-procedure*.

gdu-add-color-selection-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *list-height*: integer, *colors*: sequence, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a color picker control where you specify *colors* as a sequence of symbols that are the colors to appear in the dialog. Also adds the control to the dialog with an anchor position when used in a resizable dialog, adds the controls in a tab frame, and specifies *list-height*, *conclude-immediately*, and *validation-procedure*.

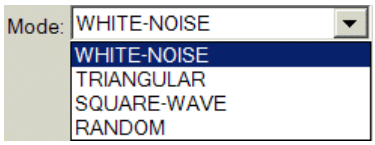
Combo Box

gdu-add-combo-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *editable-combobox*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a combo box control to the dialog, including a prompt label. Set *editable-combobox* to true to allow the user to edit the value in the text field of the combo box. Specify the *list-of-choices* as a sequence of text values to display in the combo box. Alternatively, specify *list-of-choices-procedure-name* as a procedure name whose return values provide the list of choices for the combo box. The signature of the procedure is:

my-list-of-choices-procedure
(*target*: class item, *attribute-name*: symbol, *win*: class g2-window)
–> *list-of-choices*: sequence

If neither a list of choices nor a procedure is specified, and the attribute is a symbol, the API attempts to retrieve the list of choices from the attribute description in the class definition. For example:



See the Dynamic Dialog Specification example in `gdu-demo.kb`.

gdu-add-combo-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *editable-combobox*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a combo box control to the dialog in a tab frame.

gdu-add-combo-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *editable-combobox*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a combo box control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-combo-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *editable-combobox*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *list-height*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Specifies *list-height*, which is the number of visible rows to display in the combo box. By default, the combo box shows all choices. If the number of visible rows exceeds the number of choices, the combo box has scroll bars. Adds the control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-combo-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *editable-combobox*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *localize-list-of-choices*: truth-value, *list-height*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Set *localize-list-of-choices* to true to provide localized text for the list of choices. For details, see "Localizing Dialogs" on page 23.

Adds the control to the dialog in a tab frame, and specifies *list-height*, *conclude-immediately*, and *validation-procedure*.

gdu-add-combo-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *editable-combobox*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *localize-list-of-choices*: truth-value, *list-height*: integer, *dropdown-width*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Specifies *dropdown-width*, which is the width of the combo box in dialog units. Adds the control to the dialog in a tab frame, and specifies *list-height*, *localize-list-of-choices*, *conclude-immediately*, and *validation-procedure*.

gdu-add-combo-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *editable-combobox*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *localize-list-of-choices*: truth-value, *list-height*: integer, *dropdown-width*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a combo box control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *localize-list-of-choices*, *list-height*, *dropdown-width*, *conclude-immediately* and *validation-procedure*.

Detail Button

gdu-add-detail-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *button-label*: text, *button-icon*: item-or-value, *button-action-procedure*: symbol, *user-data*: structure, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a control that consists of a label and a button for displaying dialog details. Specifies an anchor position when used in resizable dialogs. Specifies *conclude-immediately* and *validation-procedure*. Also adds the control in a tab frame. For a description of the *anchor* argument, see gdu-add-calendar-box-control.

Specify *button-label* to be the label text for the button if *button-icon* is not specified; the default is to display a button with ellipses (...). Specify *button-icon* to use an icon in place of the button text, in which case *button-text* is a tooltip for the button. Specify *button-action-procedure* to be the procedure to call when the user clicks the action button. The syntax of the procedure is:

button-action-procedure
(*dlg*: class gdu-dialog-instance, *text-box-control*: structure, *other-values*: sequence, *target*: class item, *user-data*:structure, *win*: class g2-window)
–> *text*: text, *modify*: truth-value

The button activation procedure returns a text, which is the text to appear in the text box when *modify* is true.

gdu-add-detail-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *button-label*: text, *button-icon*: item-or-value, *button-action-procedure*: symbol, *user-data*: structure, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a detail button control to a dialog that is not resizable. Specifies *conclude-immediately* and *validation-procedure*. Also adds the control in a tab frame.

gdu-add-detail-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *button-label*: text, *button-icon*: item-or-value, *button-action-procedure*: symbol, *user-data*: structure, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a detail button control to a dialog that is not resizable. Also adds the control in a tab frame.

gdu-add-detail-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *button-action-procedure*: symbol, *user-data*: structure, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a detail button control to a dialog that is not resizable, using the default button text and icon. Also adds the control in a tab frame.

gdu-add-detail-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *button-action-procedure*: symbol, *user-data*: structure, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a detail button control to a dialog that is not resizable, using the default button text and icon.

Directory Selection

gdu-add-directory-selection-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *client-side*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a directory selection control to the dialog. Specify *client-side* to determine whether to browse the client or the server. Specifies an anchor position when used in a resizable dialog. Also specifies *conclude-immediately* and *validation-procedure*. For a description of the *anchor* argument, see gdu-add-calendar-box-control.

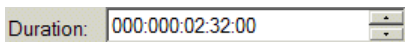
gdu-add-directory-selection-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *client-side*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a dialog selection control to the dialog in a tab frame with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*. For a description of the *anchor* argument, see gdu-add-calendar-box-control.

Duration

gdu-add-duration-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a duration control to the dialog, including a prompt label. The format for the duration is www:ddd:hh:mm:ss for weeks, days, hours, minutes, and seconds. For example:



See the Many Controls example in `gdu-demo.kb`.

gdu-add-duration-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a duration control to the dialog in a tab frame.

gdu-add-duration-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

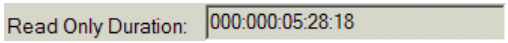
Specifies *conclude-immediately* and *validation-procedure*.

gdu-add-duration-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a duration control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-read-only-duration-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a read-only duration control to the dialog, including a prompt label. For example:



See the Many Controls example in `gdu-demo.kb`.

gdu-add-read-only-duration-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a read-only duration control to the dialog in a tab frame.

gdu-add-read-only-duration-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *anchor*: value, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a read-only duration control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

File Selection

gdu-add-file-selection-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *file-extension*: text, *filters*: sequence, *only-existing-files*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds to the dialog a text box and push button control, including a prompt label, which displays a file selection dialog when the user clicks the button. For example:

Specify:

- *file-extension* — The default file extension to use if none is supplied by the user. Specify the file extension without the "." (dot), for example, "kb".
- *filters* — A sequence of choices for the file type combo box, where each choice has the form (sequence (*label*, *wildcard*). The default value is:

  ```
  sequence(sequence("All Files (*.*)" "*.*"))
  ```

  which displays a single filter label in the combo box and allows files of any type, using the *.* wildcard:

  ```
  All Files (*.*)
  ```

  For example, to create a filter that allows `.kb` files and all files, the filter specification would look like this:

  ```
  sequence(sequence("KBs (*.kb)" "*.kb")
           sequence("All Files (*.*)" "*.*"))
  ```

- *only-existing-files* — Whether to allow the user to select only existing files (true) or whether the user can create new files (false).

You can localize the dialog title text, the OK button text, and the Cancel button text of the file selection dialog. If the file exist and the file selection dialog

needs to ask if it can overwrite the file, you can localize the confirmation dialog title text and label text. For details, see "Localizing Dialogs" on page 23.

For example:

```
call gdu-add-file-selection-control
    (dlg, the symbol file, the symbol file, true, true, "txt",
    sequence (sequence ("All Files" , "*.*" ),
                    sequence("CSV Files (.csv)", "*.csv"),
                    sequence("Text Files (.txt)", "*.txt")),
    false, 1, 0, 1, 1);
```

See the Dynamic Dialog Specification example in `gdu-demo.kb`.

gdu-add-file-selection-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *file-extension*: text, *filters*: sequence, *only-existing-files*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a file selection control to the dialog in a tab frame.

gdu-add-file-selection-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *file-extension*: text, *filters*: sequence, *only-existing-files*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a file selection control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

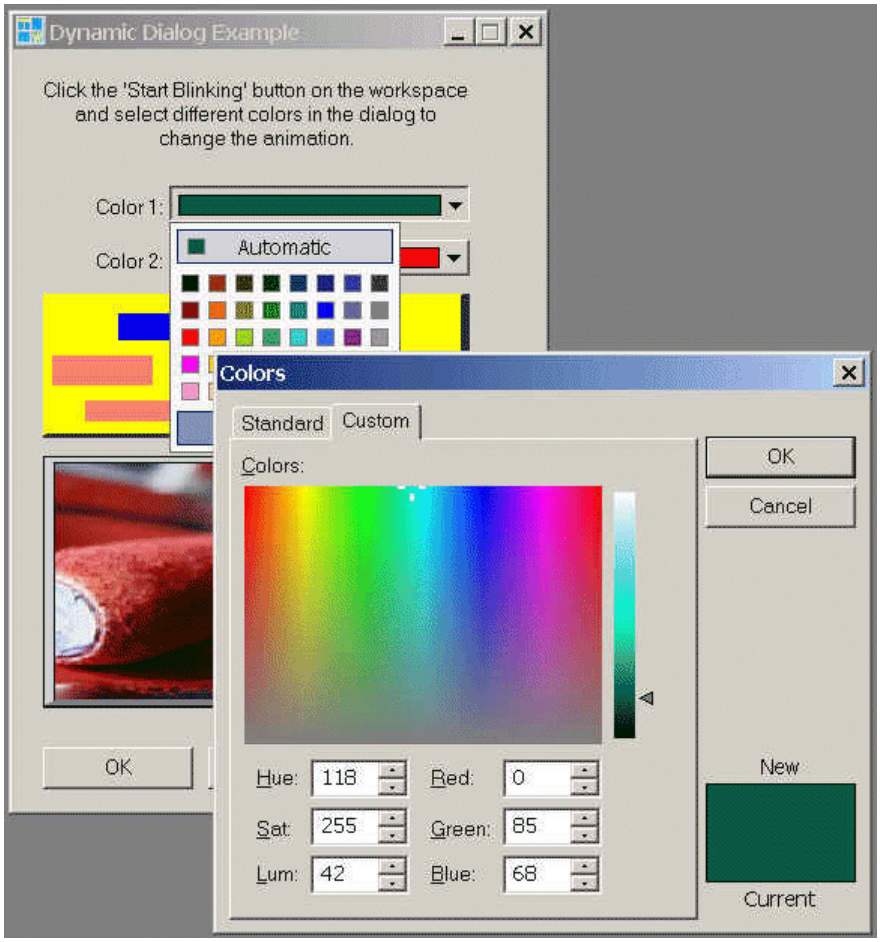
gdu-add-file-selection-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *file-extension*: text, *filters*: sequence, *only-existing-files*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a file selection control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*. For a description of the *anchor* argument, see gdu-add-calendar-box-control.

Full Color Picker

gdu-add-full-color-selection-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a 24-bit color selection control to the dialog, including a label, and specifies *conclude-immediately* and *validation-procedure*. For example:



See the Many Controls example in `gdu-demo.kb`.

gdu-add-full-color-selection-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a 24-bit color selection control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-full-color-selection-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a 24-bit color selection control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*. For a description of the *anchor* argument, see gdu-add-calendar-box-control.

G2 Editor

A read only text field is displayed in the dialog, a "..." is added next to it to launch the G2 editor and an optional button can be added before (or above if height > 1) to launch a custom procedure to insert for example G2 syntax templates that the user can then modify.

gdu-add-g2-editor-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *optional-button-label*: text, *optional-button-icon*: item-or-value, *optional-button-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a read-only text field to a dialog with a button next to the text field that launches the G2 text editor. Specify *optional-button-label*, *optional-button-icon*, and *optional-button-procedure* to add an optional button before the text field, or above it if the height is greater than 1, which executes a custom procedure that can be used to insert G2 syntax templates that the user can them modify, for example.

The signature of the *optional-button-procedure* is same as for the detail button control. It should return true and the updated text/template text to insert into the read-only text field or false and any text to cancel.

my-optional-button-procedure
(*dlg*: class gdu-dialog-instance, *text-box-control*: structure, *other-values*: sequence, *target*: class item, *user-data*: structure, *win*: class g2-window)
–> *text*: text, *status*: truth-value

gdu-add-g2-editor-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *optional-button-label*: text, *optional-button-icon*: item-or-value, *optional-button-procedure*: symbol, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a read-only text field to a dialog in a tab frame with a button next to the text field that launches the G2 text editor. You can also specify *conclude-immediately*, *validation-procedure*, and *anchor*.

Grid View

gdu-add-grid-view-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *use-row-header*: truth-value, *use-column-header*: truth-value, *allow-sort-rows*: truth-value, *columns*: sequence, *rows*: sequence, *value-formatting-procedure*: symbol, *value-unformatting-procedure*: symbol, *selection-changed-procedure*: symbol, *event-notification-procedure*: symbol, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *action-buttons*: sequence, *action-buttons-initially-enabled*: sequence, *action-button-width*: integer, *action-button-height*: integer, *action-button-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a grid view control to the dialog. The G2 application can store the data by using a variety of data structures. Therefore, this API uses a formatting procedure to convert the values retrieved by gdu-get-target-attribute-value into a format that is suitable for the grid view and assigns a logical ID to each. GDU provides four pre-defined formatting procedures, or you can implement your own. The API also provides selection changed and event notification callback procedures. You can provide a sequence of actions buttons for the grid view and an associated procedure.

Specify these arguments:

- *use-cell-width* — Set to true to make the width of the grid view span the entire width of the cell; otherwise, it spans only the width of the area for controls.
- *use-row-header* — Set to true to show the row header.
- *use-column-header* — Set to true to show the column header.
- *allow-sort-rows* — Set to true to allow the rows to be sorted.

- Set *columns* to a sequence of structures that corresponds with the columns attribute of the control-value of the grid-view control. The options for default-cell-type are: integer, quantity, text-box, check-box, combo-box, color-picker, duration, calendar, time-of-day, spinner, and image. The options for width are: header-width and text-width.

  For details, see the description of grid-view in Chapter 47, "Custom Windows Dialogs" on page 1431 in the *G2 Reference Manual*.

- Set *rows* to a sequence of structures that corresponds with the rows attribute of the control-value of the grid-view control. The options for cell-type are the same as for *columns*.

  For details, see the description of grid-view in Chapter 47, "Custom Windows Dialogs" on page 1431 the *G2 Reference Manual*.

- *value-formatting-procedure* — A procedure that is called to transfer the values returned by gdu-get-target-attribute-value to the data format expected by the grid view. The syntax of the procedure is:

  ```
  my-grid-view-data-value-formatter
      (target: class item, unformatted-values: sequence,
      dlg: class gdu-dialog-instance, control: structure,
      win: class g2-window)
      -> formatted-values: sequence, number-of-rows: integer
  ```

  Returns the values formatted as required by the rows attribute of the control-value structure of the grid-view control, and the number of rows, which is also the greatest row-id.

  GDU includes the following pre-defined formatting procedures, which format values stored in the attribute or as computed in the specified format by gdu-get-target-attribute-value:

  - gdu-grid-view-format-sequence-of-structures-data-values-to-cell-values
  - gdu-grid-view-format-sequence-of-sequences-data-values-to-cell-values
  - gdu-grid-view-format-sequence-data-values-to-cell-values
  - gdu-grid-view-format-structure-of-sequences-data-value-to-cell-values
  - gdu-grid-view-format-sequence-of-objects-data-values-to-cell-values

  These procedures are identical to those defined for a tabular view, except for the names. For a description, see "Tabular View" on page 88.

- *value-unformatting-procedure* — A procedure that is called to transform the values from the grid view cell format to the attribute format on the target object. The syntax of the procedure is:

  ```
  my-grid-view-format-cell-values-to-sequence-of-structures-data-values
      (target: class item, new-value: structure,
      dlg: class gdu-dialog-instance, control: structure,
      win: class g2-window)
      -> value: value
  ```

  GDU includes the following pre-defined unformatting procedures:

  - gdu-grid-view-format-cell-values-to-sequence-of-structures-data-values
  - gdu-grid-view-format-cell-values-to-sequence-of-sequences-data-values
  - gdu-grid-view-format-cell-values-to-sequence-data-values
  - gdu-grid-view-format-cell-values-to-structure-of-sequences-data-values
  - gdu-grid-view-format-cell-values-to-sequence-of-object-data-values

- *selection-changed-procedure* — A procedure that is called when the user selects rows of the grid view. The signature of this procedure is:

  ```
  my-grid-view-selection-procedure
      (target: class item, dlg: class gdu-dialog-instance,
      control-id: value, selection: structure
      win: class g2-window)
  ```

  GDU includes the following pre-defined selection changed procedure for handling enable/disable and delete/remove row (*action-id* = the symbol grtl-icon-remove-row or the symbol grtl-delete-row-action-button) action buttons:

  - gdu-grid-view-default-selection-notification-procedure

- *event-notification-procedure* — A procedure that is called when mouse click and keyboard events occur in the grid view. The signature of this procedure is:

  ```
  my-grid-view-event-notification-procedure
      (target: class item, dlg: class gdu-dialog-instance,
      control-id: symbol, control: structure, event: symbol, info: structure,
      win: class g2-window)
  ```

  For a description of the events, see the description of the "Generic Dialog Callback" in Chapter 47, "Custom Windows Dialogs" in the *G2 Reference Manual*.

- *validation-procedure* — A procedure with the following signature that is called when a value in the grid view changes:

  ```
  my-grid-view-validation-procedure
      (dlg: class gdu-dialog-instance, control: structure,
      new-control-value: structure, other-values: sequence,
      target: class item, Win: class g2-window)
      –> valid: truth-value
  ```

- *action-buttons* — A list of symbolic values corresponding to actions associated with the grid view. A push button is displayed for each action in a row above the grid view. If the symbol name corresponds to a class definition, the icon of the class definition is displayed. The grtl-icons module provides the grl-icon-insert-row and grtl-icon-remove-row action buttons to add or remove the selected row.

- *action-button-initially-enabled* — A sequence of truth-values of the same length as the *action-buttons* argument. Each truth-value specifies whether the action button with the same index is initially enabled or disabled.

- *action-button-width* — The width of each action button, in pixels.

- *action-button-height* — The height of each action button, in pixels.

- *action-button-procedure* — A procedure to call when the user clicks an action button in the grid view. The syntax of the procedure is:

  ```
  my-grid-view-activation-procedure
      (target: class item, dlg: class gdu-dialog-instance,
      action-id: symbol, grid-view-control-id: symbol,
      selection: structure, win: class g2-window)
  ```

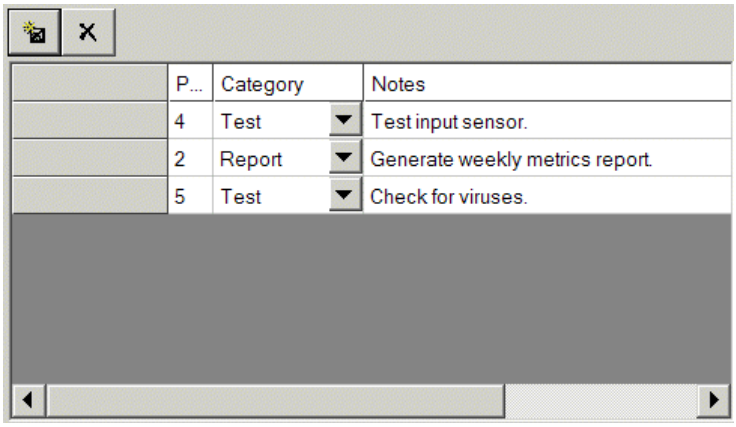
  GDU includes the following pre-defined action button procedure for handling add row (*action-id* = the symbol grtl-icon-insert-row) and remove row (*action-id* = the symbol grtl-icon-remove-row or the symbol grtl-delete-row-action-button) action buttons:

  – gdu-grid-view-default-button-activation-procedure

  To collect the specification of a new row, the default implementation calls the following method, which returns a new row specification as defined by the grid-view. The default implementation computes the row specification from the column specifications, which requires that default values be used. The row header is an empty string. The signature of this method is:

  ```
  gdu-get-target-new-grid-view-row
      (target: class item, dlg: class gdu-dialog-instance,
      grid-view-control-id: symbol, new-row-index: integer,
      win: class g2-window)
      –> row-specification: structure
  ```

For example:



See the Grid View example in `gdu-demo.kb`.

gdu-add-grid-view-control (
dlg: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol,
initially-visible: truth-value, *initially-enabled*: truth-value,
use-cell-width: truth-value, *use-row-header*: truth-value,
use-column-header: truth-value, *allow-sort-rows*: truth-value,
columns: sequence, *rows*: sequence,
value-formatting-procedure: symbol, *value-unformatting-procedure*: symbol
selection-changed-procedure: symbol,
action-buttons: sequence, *action-button-initially-enabled*: sequence,
action-button-width: integer, *action-button-height*: integer,
action-button-procedure: symbol,
grid-x: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a grid view control to the dialog specifying a *selection-changed-procedure* and action buttons.

gdu-add-grid-view-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol,
initially-visible: truth-value, *initially-enabled*: truth-value,
use-cell-width: truth-value, *use-row-header*: truth-value,
use-column-header: truth-value, *allow-sort-rows*: truth-value,
columns: sequence, *rows*: sequence,
value-formatting-procedure: symbol, *value-unformatting-procedure*: symbol,
selection-changed-procedure: symbol,
action-buttons: sequence, *action-buttons-initially-enabled*: sequence,
action-button-width: integer, *action-button-height*: integer,
action-button-procedure: symbol, *grid-x*: integer, *grid-y*: integer,
grid-x-span: integer, *grid-y-span*: integer,
tab-control: value, *tab-name*: text)

Adds a grid view control to the dialog in a tab frame, specifying a *selection-changed-procedure* and action buttons.

gdu-add-grid-view-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol,
initially-visible: truth-value, *initially-enabled*: truth-value,
use-cell-width: truth-value, *use-row-header*: truth-value,
use-column-header: truth-value, *allow-sort-rows*: truth-value,
columns: sequence, *rows*: sequence,
value-formatting-procedure: symbol, *value-unformatting-procedure*: symbol,
selection-changed-procedure: symbol, *event-notification-procedure*: symbol,
conclude-immediately: truth-value, *validation-procedure*: symbol,
anchor: value, *grid-x*: integer, *grid-y*: integer,
grid-x-span: integer, *grid-y-span*: integer,
tab-control: value, *tab-name*: text)

Adds a grid view control to the dialog in a tab frame, specifying a *selection-changed-procedure* and an *event-notification-procedure* and no action buttons.

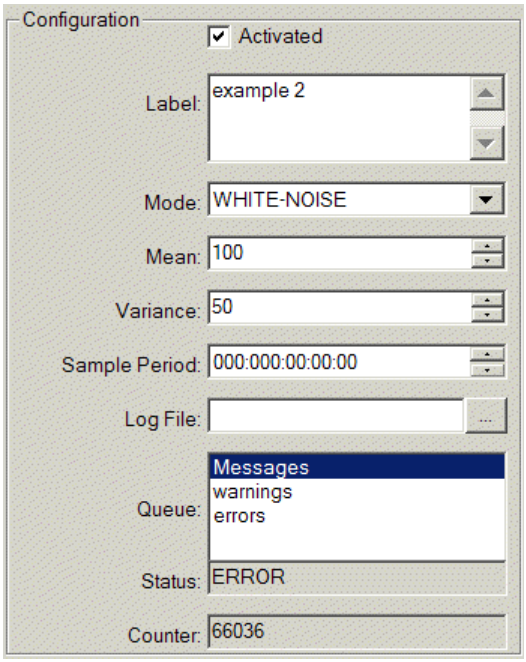
gdu-add-grid-view-control (
dlg: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol,
initially-visible: truth-value, *initially-enabled*: truth-value,
use-cell-width: truth-value, *use-row-header*: truth-value,
use-column-header: truth-value, *allow-sort-rows*: truth-value ,
columns: sequence, *rows*: sequence,
value-formatting-procedure: symbol, *value-unformatting-procedure*: symbol
selection-changed-procedure: symbol,
action-buttons: sequence, *action-button-initially-enabled*: sequence,
action-button-width: integer, *action-button-height*: integer,
action-button-procedure: symbol,
grid-x: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer,
tab-control: value, *tab-name*: text)

Adds a grid view control to the dialog in a tab frame specifying a *selection-changed-procedure*.

Group

gdu-add-group-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *label*: text, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a group control to the dialog, which puts a box around a group of controls in a dialog. You specify the *label* to display as the group label. For example, here is a group control whose label is Configuration:



See the Dynamic Dialog Specification example in `gdu-demo.kb`.

gdu-add-group-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *label*: text, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a group control to the dialog in a tab frame.

gdu-add-group-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *label*: text, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a group control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame.

Image

gdu-add-image-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *icon*: item-or-value, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds an image control to the dialog in a tab frame. For example:

See the Dynamic Dialog Specification example in `gdu-demo.kb`.

gdu-add-image-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *icon*: item-or-value, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds an image control to the dialog with an anchor position when used in a resizable dialog. Also adds the control to the dialog in a tab frame. For a description of the *anchor* argument, see gdu-add-calendar-box-control.

Instance Selection

Added the following APIs to build the list of instances and add the combobox control. The instances listed in the combobox maybe limited to those found on a specific workspace. unspecified-value specifies the value that can be used to specify that none is selected and class-names argument should contain a list of symbols naming class names.

gdu-add-instance-selection-combo-box-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *editable-combobox*: truth-value, *unspecified-value*: text, *class-names*: sequence, *upon-workspace*: item-or-value, *named-items-only*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a combo box containing a list of instances of the specified classes, where *unspecified-value* specifies the value to be used when none is selected and *class-names* is a list of symbols naming class names whose instances should be created. Alternatively, specify *upon-workspace* to specify a workspace whose items should appear in the combo box and *named-items-only* to specify that only named items on the workspace should appear.

gdu-add-instance-selection-combo-box-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *editable-combobox*: truth-value, *unspecified-value*: text, *class-names*: sequence, *upon-workspace*: item-or-value, *named-items-only*: truth-value, *list-height*: integer, *dropdown-width*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a combo box control to a tab frame with a list of instances from which to choose, where you can specify the *list-height* and *dropdown-width* of the combo box. You can also specify *conclude-immediately*, *validation-procedure*, and *anchor*.

Label

gdu-add-label-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *label*: text, *initially-visible*: truth-value, *initially-enabled*: truth-value, *add-colon-upon-localization*: truth-value, *use-cell-width*: truth-value, *alignment*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a label control to the dialog. Set *add-colon-upon-localization* to true to add a colon to the label text when localizing the text. Set *use-cell-width* to true to cause the label to extend the full width of the cell in the grid. Specify *alignment* as left, right, and center to align the label within the grid. Note that many controls specify prompt labels in addition to the dialog control, although some do not. For example, here is a label control that is centered in the dialog across all columns:

You can provide localized text for the label text by referring directly to the control ID as the key. For details, see "Localizing Dialogs" on page 23.

See the Many Controls example in `gdu-demo.kb`.

gdu-add-label-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *label*: text,
initially-visible: truth-value, *initially-enabled*: truth-value,
add-colon-upon-localization: truth-value, *use-cell-width*: truth-value,
alignment: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer,
grid-y-span: integer, *tab-control*: value, *tab-name*: text)

Adds a label control to the dialog in a tab frame.

gdu-add-label-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *label*: text,
initially-visible: truth-value, *initially-enabled*: truth-value,
add-colon-upon-localization: truth-value, *use-cell-width*: truth-value,
alignment: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer,
grid-x-span: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a label control to the dialog with an anchor position when used in a resizable dialog. Also adds the control to the dialog in a tab frame. For a description of the *anchor* argument, see gdu-add-calendar-box-control.

List Box

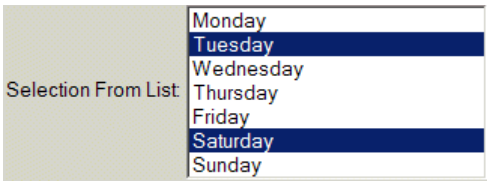
gdu-add-list-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol,
initially-visible: truth-value, *initially-enabled*: truth-value,
multiple-selection: truth-value, *extended-selection*: truth-value,
list-of-choices: sequence, *list-of-choices-procedure-name*: symbol,
grid-x: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a list box control to the dialog, including a prompt label. Set *multiple-selection to* true to allow the user to select of any number of elements in the list box by single clicking the element. Clicking a selected element deselects it. Set *extended-selection* to true to allow the user to extend the selection to include multiple entries. Selecting an element deselects the currently selected elements. Holding down the SHIFT key and clicking an element selects a series of elements. Holding down the CTRL key and clicking an element adds or removes individual elements to or from the selection.

Specify *list-of-choices* as a sequence of text values to display in the list box. The *attribute-name* specifies the initially selected items in the list. Alternatively, specify *list-of-choices-procedure-name* as a procedure name whose return values provide the list of choices for the list box. The signature of the procedure is:

my-list-of-choices-procedure
(*target*: class item, *attribute-name*: symbol, *win*: class g2-window)
–> *list-of-choices*: sequence

If neither a list of choices nor a procedure is specified, and the attribute is a symbol, the API attempts to retrieve the list of choices from the attribute description in the class definition. For example, here is a multiple selection list box control:



See the Multiple Selection List Box example in `gdu-demo.kb`.

gdu-add-list-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *multiple-selection*: truth-value, *extended-selection*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a list box control to the dialog in a tab frame.

gdu-add-list-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *multiple-selection*: truth-value, *extended-selection*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a list box control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-list-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *multiple-selection*: truth-value, *extended-selection*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *localize-list-of-choices*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Set *localize-list-of-choices* to true to provide localized text for the list of choices. For details, see "Localizing Dialogs" on page 23. Specifies *conclude-immediately* and *validation-procedure*.

gdu-add-list-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *multiple-selection*: truth-value, *extended-selection*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *localize-list-of-choices*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

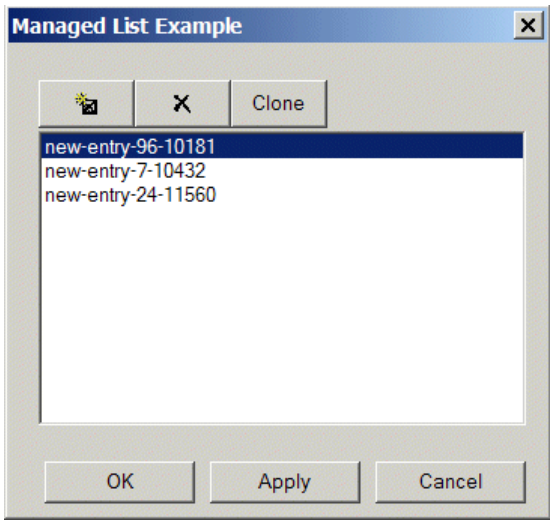
Allows localization of the list of choices by setting *localize-list-of-choices* to true. Adds the control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-list-box-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *multiple-selection*: truth-value, *extended-selection*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *localize-list-of-choices*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a list box control to the dialog with an anchor position when used in a resizable dialog. Also allows localization of the list of choices by setting *localize-list-of-choices* to true, and adds the control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*. For a description of the *anchor* argument, see gdu-add-calendar-box-control.

gdu-add-list-box-control-with-action-buttons
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *action-buttons*: sequence, *action-button-width*: integer, *action-button-height*: integer, *initial-list-content*: sequence, *action-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a list box control to the dialog with push button controls for interacting with elements in the list. The list box allows multiple and extended selection. For example:



Specify *action-buttons* as a sequence of symbols of class names that define icons to display above the list box. Specify *action-button-width* and *action-button-height* as the width and height of each action button, in pixels. Specify *initial-list-contents* as a sequence of items to display in the list box when the dialog initially opens. Specify *action-procedure* as the procedure to call when the user clicks one of the action buttons. This procedure is used as the action-procedure in the control structure for each push button control that gets created. The signature of this procedure is:

my-activation-procedure
(*target*: class item, *dlg*: class gdu-dialog-definition, *action-id*: symbol, *list-control-id*: symbol, *list-content*: sequence, *list-selection*: sequence, *win*: class g2-window)

See the List Box with Action Buttons example in `gdu-demo.kb`.

gdu-add-list-box-control-with-action-buttons
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *action-buttons*: sequence, *action-button-width*: integer, *action-button-height*: integer, *initial-list-content*: sequence, *action-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a list box control with push buttons to the dialog in a tab frame.

gdu-add-list-box-control-with-action-buttons
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *action-buttons*: sequence, *action-buttons-enable*: sequence, *action-button-width*: integer, *action-button-height*: integer, *initial-list-content*: sequence, *action-procedure*: symbol, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

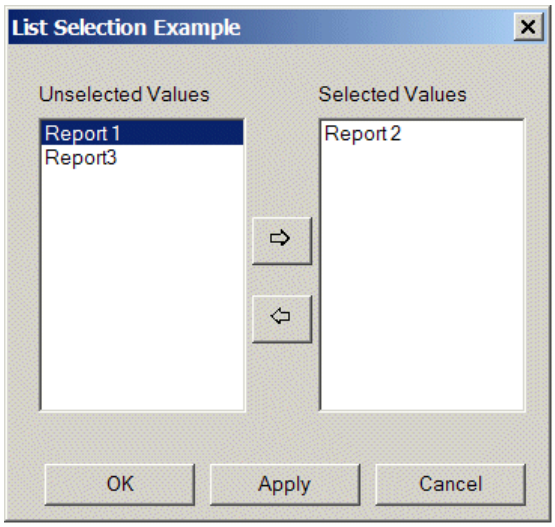
Adds a list box control with push buttons to the dialog where you specify *action-buttons-enable* as a sequence of truth-values to determine whether the buttons are initially enabled or disabled. Also specifies an anchor position when used in a resizable dialog, adds the control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*. For a description of the *anchor* argument, see gdu-add-calendar-box-control.

gdu-add-single-selection-list-box-control-with-action-buttons
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *action-buttons*: sequence, *action-buttons-enable*: sequence, *action-button-width*: integer, *action-button-height*: integer, *initial-list-content*: sequence, *action-procedure*: symbol, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a list box control with push buttons to the dialog, where the list box does not allow multiple or extended selection. You specify *action-buttons-enable* as a sequence of truth-values to determine whether the buttons are initially enabled or disabled. Also specifies an anchor position when used in a resizable dialog, adds the control to the dialog in a tab frame, specifies *conclude-immediately* and *validation-procedure*. For a description of the *anchor* argument, see gdu-add-calendar-box-control. The arguments to this procedure are identical to the arguments to gdu-add-list-box-control-with-action-buttons.

gdu-add-list-selection-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *label-for-list-on-left*: text, *label-for-list-on-right*: text, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds two list box controls with arrows for moving items between the two lists and labels above each list. For example:



Specify *label-for-list-on-left* and *label-for-list-on-right* as the labels to appear above each list.

See the List Box Selection example in `gdu-demo.kb`.

gdu-add-list-selection-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *label-for-list-on-left*: text, *label-for-list-on-right*: text, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

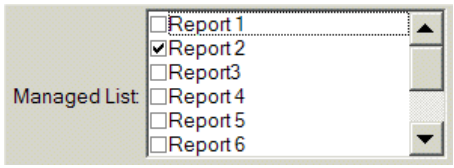
Adds a list selection control to the dialog in a tab frame.

gdu-add-list-selection-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *list-of-choices:* sequence, *list-of-choices-procedure-name*: symbol, *label-for-list-on-left*: text, *label-for-list-on-right*: text, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a list selection control to the dialog at an anchor position when used in a resizable dialog. Also adds the control in a tab frame. For a description of the *anchor* argument, see gdu-add-calendar-box-control.

gdu-add-checkable-list-box-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *multiple-selection*: truth-value, *extended-selection*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *localize-list-of-choices*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a checkable list box control to the dialog, including a prompt label. For example:



gdu-add-checkable-list-box-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *multiple-selection*: truth-value, *extended-selection*: truth-value, *list-of-choices*: sequence, *list-of-choices-procedure-name*: symbol, *localize-list-of-choices*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a checkable list box control to the dialog in a tab frame.

gdu-update-content-of-list-box-control
(*dlg*: gdu-dialog-definition, *list-control-id*: symbol, *new-content*: sequence, *new-selection*: sequence)

Updates the contents of a list box control with new elements. Specify the *list-control-id* in the given dialog to update, the *new-content* as the new sequence of text elements to appear in the list, and *new-selection* as the new sequence of text elements to select.

See gdu-managed-list-box-activation-procedure in the List Box with Action Buttons example in `gdu-demo.kb`.

Masked Edit

gdu-add-masked-edit-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *literal*: text, *mask*: text, *format-value-procedure*: symbol, *unformat-value-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a masked edit control to the dialog, including a prompt label, which restricts what the user can enter in a text box. The *attribute-value* determines the initial value when the control is displayed, as well as the character to enter when the user presses the Backspace key. You might be required to implement a custom get and set method to format the attribute value appropriately for the masked edit.

Specify *literal* as the literal format for the data, where an underscore ("_") indicates where the user can enter data. For example, to specify a phone number, specify *literal* as: "(___) ___-____").

Specify *mask* the position and type of the values the user can enter and any fixed text. Use these characters to specify the type of allowable text:

0 — Numeric (0-9)

9 — Numeric (0-9) or space (' ')

— Numeric (0-9) or space (' ') or ('+') or ('+')

L — Alpha (a-Z)

? — Alpha (a-Z) or space (' ')

A — Alpha numeric (0-9 and a-Z)

a — Alpha numeric (0-9 and a-Z) or space (' ')

& — All print character only

H — Hex digit (0-9 and A-F)

X — Hex digit (0-9 and A-F) and space (' ')

> — Forces characters to upper case (A-Z)

< — Forces characters to lower case (a-z)

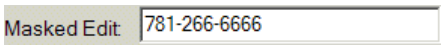
All other characters are considered fixed text.

Specify *format-value-procedure* as the procedure to call when the user enters values into the masked edit control to format the values enters. Specify

unformat-value-procedure as the procedure to call when the user presses the Backspace key to unformat text. The syntax of both procedures is:

```
my-format-value-procedure
    (control: structure, val: value)
    -> formatted-text: text
```

For example:



See the Many Controls example in `gdu-demo.kb`.

gdu-add-masked-edit-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *literal*: text, *mask*: text, *format-value-procedure*: symbol, *unformat-value-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a masked edit control to the dialog in a tab frame.

gdu-add-masked-edit-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *literal*: text, *mask*: text, *format-value-procedure*: symbol, *unformat-value-procedure*: symbol, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a masked edit control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-masked-edit-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *literal*: text, *mask*: text, *format-value-procedure*: symbol, *unformat-value-procedure*: symbol, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a masked edit control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

Progress Bar

gdu-add-progress-bar-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *is-smooth*: truth-value, *is-vertical*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a progress bar control to the dialog, including a prompt label. Specify *min-value*, *max-value*, and *increment* as the minimum and maximum values and the amount to increment. Set *is-smooth* to true to show the increments as smooth steps, or set to false to show the increments as discrete steps. Set *is-vertical* to true to make the progress bar vertical, or set to false to make it horizontal. For example, here is a progress bar that increments by using discrete steps:

gdu-add-progress-bar-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *is-smooth*: truth-value, *is-vertical*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a progress bar control to the dialog in a tab frame.

gdu-add-progress-bar-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *is-smooth*: truth-value, *is-vertical*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a progress bar control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

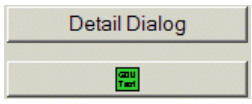
gdu-add-progress-bar-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *is-smooth*: truth-value, *is-vertical*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a progress bar control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

Push Button

gdu-add-push-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *label*: text, *icon-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *activation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a push button to the dialog. Specify *label* as the text to display in the push button, and specify *icon-name* as the class name whose icon description to use as an icon in the push button instead of the label. Set *use-cell-width* to true to use the full width of the grid cell for the control, which left-aligns the control in the cell. For example, here are two push button controls, one with text and one with an icon:



You can provide localized text for the push button label by referring directly to the control ID as the key. For details, see "Localizing Dialogs" on page 23.

gdu-add-push-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *label*: text, *icon-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *activation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Displays a push button control in a dialog. Specify *activation-procedure* as the procedure to call when the user clicks the push button. The activation procedure must have this syntax:

activation-procedure
(*dlg*: class gdu-dialog-instance, *control*: structure, *other-values*: sequence, *target*: class item, *win*: class g2-window)"

See the Many Controls example in `gdu-demo.kb`.

gdu-add-push-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *label*: text, *icon-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *activation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a push button control to the dialog in a tab frame.

gdu-add-push-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *label*: text, *icon-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *activation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds the push button control to the dialog in a tab frame and specifies *use-cell-width*.

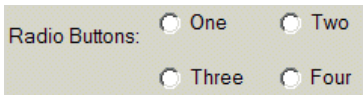
gdu-add-push-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *label*: text, *icon-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *activation-procedure*: symbol, *anchor*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a push button control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *use-cell-width* and *activation-procedure*.

Radio Button

gdu-add-radio-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *on-values*: sequence, *nb-of-radio-buttons-per-cell*: integer, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a radio button control to the dialog, including a prompt label. Specify *on-values* as a sequence of text values to appear as radio buttons. The *attribute-name* specifies the radio button that is initially selected. Specify *nb-of-radio-buttons-per-cell* as the number of columns of radio buttons to display in the grid cell. For example, if you specify four values in *on-values* and you specify 2 radio buttons per cell, the radio buttons would be formatted like this:



You can provide localized text for the radio button text values. For details, see "Localizing Dialogs" on page 23.

See the Many Controls example in `gdu-demo.kb`.

gdu-add-radio-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *on-values*: sequence, *nb-of-radio-buttons-per-cell*: integer, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a radio button control to the dialog in a tab frame.

gdu-add-radio-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *on-values*: sequence, *nb-of-radio-buttons-per-cell*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a radio button control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-radio-button-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *on-values*: sequence, *nb-of-radio-buttons-per-cell*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a radio-button control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

Slider

gdu-add-slider-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a slider control to the dialog, including a prompt label. Specify *min-value*, *max-value*, and *increment* as the minimum and maximum values and the amount to increment. For example:

See the Many Controls example in `gdu-demo.kb`.

gdu-add-slider-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a slider control to the dialog in a tab frame.

gdu-add-slider-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a slider control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-slider-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a slider control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

Spinner

gdu-add-spinner-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: quantity, *max-value*: quantity, *increment*: quantity, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Displays a spinner control in the dialog, including a prompt label. Specify *min-value*, *max-value*, and *increment* as the minimum and maximum values and the amount to increment. The user increments values by clicking the arrows to the right of the text box or by entering values in the text box. For example:

See the Dynamic Dialog Specification example in `gdu-demo.kb`.

gdu-add-spinner-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: quantity, *max-value*: quantity, *increment*: quantity, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a spinner control to the dialog in a tab frame.

gdu-add-spinner-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: quantity, *max-value*: quantity, *increment*: quantity, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a spinner control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-spinner-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: quantity, *max-value*: quantity, *increment*: quantity, *integer-values*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a spinner control to the dialog with an anchor position when used in a resizable dialog. Specify *integer-values* to determine if the spinner accepts only integers. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-integer-spinner-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: quantity, *max-value*: quantity, *increment*: quantity, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Displays a spinner control in a dialog that only accepts integer values, regardless of the minimum, maximum, and increment.

See the Dynamic Dialog Specification example in `gdu-demo.kb`.

gdu-add-integer-spinner-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: quantity, *max-value*: quantity, *increment*: quantity, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds an integer spinner control to the dialog in a tab frame.

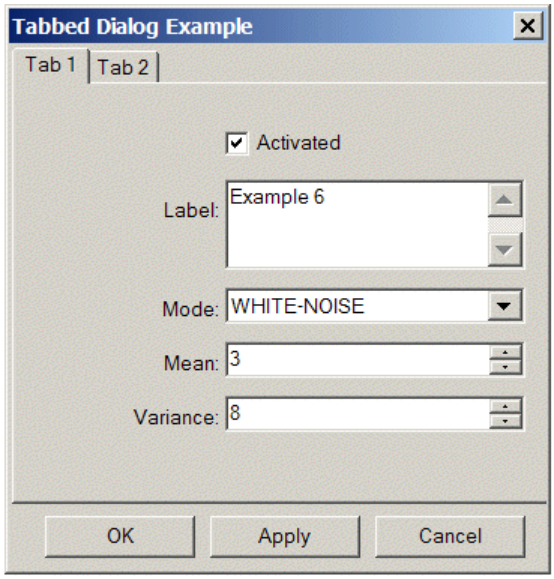
gdu-add-integer-spinner-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: quantity, *max-value*: quantity, *increment*: quantity, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a spinner control to the dialog that only accepts integer values. Also adds the control with an anchor position when used in a resizable dialog and adds the control in a tab frame.

Tab Frame

gdu-add-tab-frame-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *tabs*: sequence,
grid-x: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a tab frame control to the dialog. Specify *tabs* as a sequence of text values to appear on each tab. You add controls to each tab frame by using the APIs that provide *tab-control* and *tab-name* arguments. You specify the *control-id* of the tab frame control and the text of the tab in which the control should appear. For example, here is a dialog with two tabs:



See the Tabbed Dialogs example in `gdu-demo.kb`.

gdu-add-tab-frame-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *tabs*: sequence,
grid-x: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer,
tab-control: value, *tab-name*: text)

Adds a tab frame control to a dialog within another tab frame.

gdu-add-tab-frame-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *tabs*: sequence,
anchor: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer,
grid-y-span: integer, *tab-control*: value, *tab-name*: text)

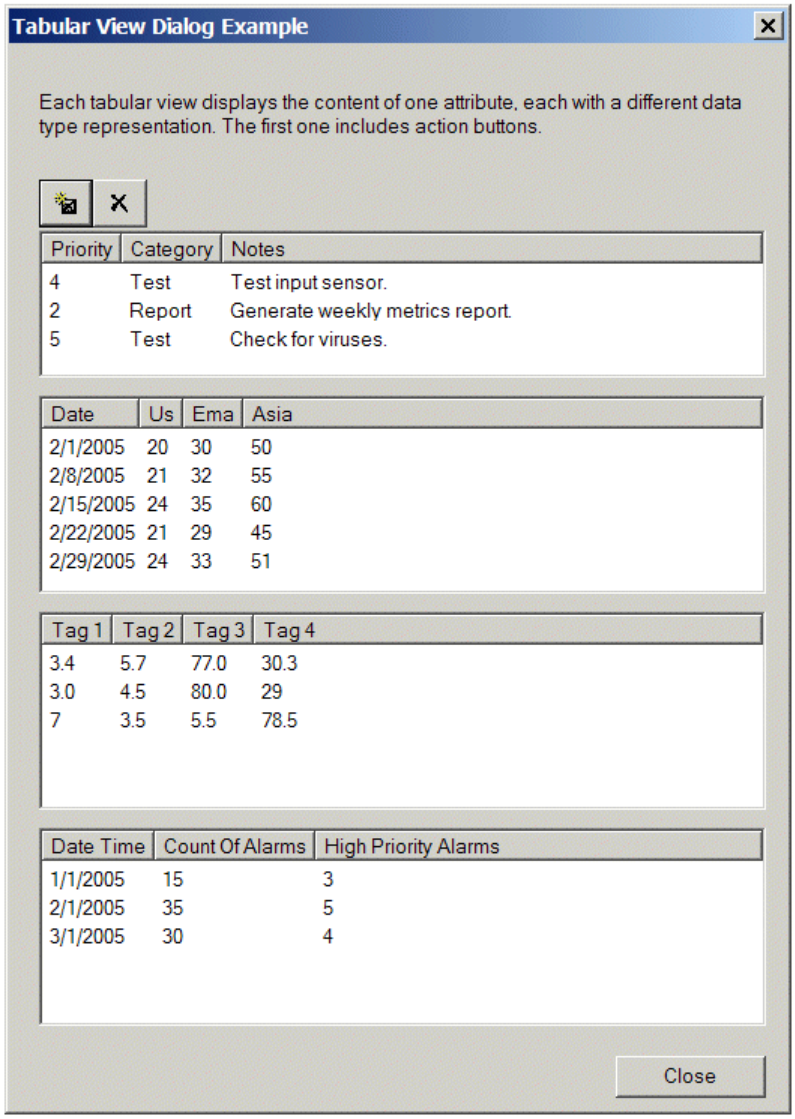
Adds a tab frame control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame.

Tabular View

gdu-add-tabular-view-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *gridlines*: truth-value, *allow-sort-rows*: truth-value, *use-cell-width*: truth-value, *columns*: sequence, *value-formatting-procedure*: symbol, *selection-changed-procedure*: symbol, *event-notification-procedure*: symbol, *use-single-selection*: truth-value, *action-buttons*: sequence, *action-buttons-initially-enabled*: sequence, *action-button-width*: integer, *action-button-height*: integer, *action-button-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a read-only tabular view control to the dialog. The G2 application can store the data by using a variety of data structures. Therefore, this API uses a formatting procedure to convert the values retrieved by gdu-get-target-attribute-value into a format that is suitable for the tabular view and assigns a logical ID to each. GDU provides various pre-defined formatting procedures, or you can implement your own. The API also provides selection changed and event notification callback procedures. You can provide a sequence of actions buttons for the tabular view and an associated procedure.

For example, this dialog includes four tabular views with buttons above the first one:



Specify these arguments:

- *gridlines* — Set to true to display gridlines.
- *allow-sort-rows* — Set to true to allow sorting of rows.
- *use-cell-width* — Set to true to make the width of the tabular view span the entire width of the grid cell; otherwise, it spans only the width of the area for controls.
- *columns* — A list of symbolic values displayed in the column header and used to extract the attributes to display from the datasets.

- *value-formatting-procedure* — A procedure that is called to transfer the values returned by gdu-get-target-attribute-value to the data format expected by the tabular view. The syntax of the procedure is:

  my-tabular-view-data-value-formatter
  (*target*: class item, *unformatted-values*: sequence,
  dlg: class gdu-dialog-instance, *control*: structure,
  win: class g2-window)
  –> *formatted-values*: sequence, *number-of-rows*: integer

  Returns the values formatted as required by the rows attribute of the control-value structure of the tabular-view control, and the number of rows, which is also the greatest row-id.

  GDU includes the following pre-defined formatting procedures, which format values stored in the attribute or as computed in the specified format by gdu-get-target-attribute-value:

  - gdu-tabular-view-sequence-of-structures-data-value-formatter
  - gdu-tabular-view-sequence-of-sequences-data-value-formatter
  - gdu-tabular-view-sequence-data-value-formatter
  - gdu-tabular-view-structure-of-sequences-data-value-formatter
  - gdu-tabular-view-sequence-of-objects-formatter

  The description of these procedures follows.

- *selection-changed-procedure* — A procedure that is called when the user selects rows of the tabular view. The signature of this procedure is:

  my-tabular-view-selection-procedure
  (*target*: class item, *dlg*: class gdu-dialog-instance,
  tabular-view-control-id: value, *selected-rows*: structure,
  win: class g2-window)

- *event-notification-procedure* — A procedure that is called when mouse click and keyboard events occur in the tabular view. The signature of this procedure is:

  my-tabular-view-event-notification-procedure
  (*target*: class item, *dlg*: class gdu-dialog-instance,
  control-id: symbol, *control*: structure, *event*: symbol, *info*: structure,
  win: class g2-window)

  For a description of the events, see the description of the "Generic Dialog Callback" in Chapter 47, "Custom Windows Dialogs" in the *G2 Reference Manual*.

- *use-single-selection* — Set to true to enable only one row in the tabular view to be selected at a time.

- *action-buttons-enable* — A sequence of truth-values corresponding to the *action-buttons* sequence indicating whether the button is initially enabled or disabled.
- *action-buttons* — A list of symbolic values corresponding to actions associated with the tabular view. A push button is displayed for each action in a row above the tabular view. If the symbol name corresponds to a class definition, the icon of the class definition is displayed.
- *action-button-width* — The width of each action button, in pixels.
- *action-button-height* — The height of each action button, in pixels.
- *action-procedure* — A procedure to call when the user clicks an action button in the tabular view. The syntax of the procedure is:

```
my-tabular-view-activation-procedure
    (target: class item, dlg: class gdu-dialog-instance,
    action-id: symbol, tabular-view-control-id: symbol,
    selected-rows: sequence, win: class g2-window)
```

See the Tabular View example in `gdu-demo.kb`.

gdu-add-tabular-view-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *gridlines*: truth-value, *allow-sort-rows*: truth-value, *use-cell-width*: truth-value, *columns*: sequence, *tabular-view-value-formatting-procedure*: symbol, *tabular-view-selection-procedure*: symbol, *use-single-selection*: truth-value, *action-buttons-enable*: sequence, *action-buttons*: sequence, *action-button-width*: integer, *action-button-height*: integer, *action-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

gdu-add-tabular-view-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *gridlines*: truth-value, *allow-sort-rows*: truth-value, *use-cell-width*: truth-value, *columns*: sequence, *tabular-view-value-formatting-procedure*: symbol, *tabular-view-selection-procedure*: symbol, *use-single-selection*: truth-value, *action-buttons-enable*: sequence, *action-buttons*: sequence, *action-button-width*: integer, *action-button-height*: integer, *action-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a tabular view control to the dialog in a tab frame.

gdu-add-tabular-view-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *gridlines*: truth-value, *allow-sort-rows*: truth-value, *use-cell-width*: truth-value, *columns*: sequence, *tabular-view-value-formatting-procedure*: symbol, *tabular-view-selection-procedure*: symbol, *use-single-selection*: truth-value, *action-buttons*: sequence, *action-buttons-enable*: sequence, *action-button-width*: integer, *action-button-height*: integer, *action-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *anchor*: value, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a tabular view control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame.

gdu-tabular-view-add-rows
(*dlg*: gdu-dialog-definition, *tabular-view-control-id*: symbol, *rows*: sequence)

Adds rows to a tabular view as a sequence of structures, where each structure has this syntax:

structure
(logical-id: *integer*,
row-values: *sequence*,
background-color: *background-color*, {optional}
text-color: *foreground-color* {optional})

For example, this code adds a single row to the tabular view:

```
call gdu-tabular-view-add-rows
     (dlg, tabular-view-control-id,
     sequence
          (structure (
               logical-id: random (50, 150),
               row-values: sequence (
                                   structure (text-value: "8"),
                                   structure (text-value: "Test [random(1,100)]"),
                                   structure (text-value: "Example [random(1,100)]")))))
```

gdu-tabular-view-remove-all-rows
(*dlg*: gdu-dialog-definition, *tabular-view-control-id*: symbol)

Removes all rows from a tabular view.

gdu-tabular-view-remove-all-selected-rows
(*dlg*: gdu-dialog-definition, *tabular-view-control-id*: symbol)

Removes all selected rows from a tabular view.

gdu-tabular-view-remove-rows
(*dlg*: gdu-dialog-definition, *tabular-view-control-id*: symbol,
logical-ids-of-rows-to-remove: sequence)

Removes the specified rows from a tabular view, where *rows* is sequence of logical-id values.

gdu-tabular-view-replace-rows
(*dlg*: gdu-dialog-definition, *tabular-view-control-id*: symbol, *rows*: sequence)

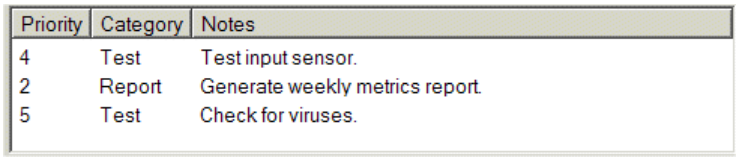
Replaces the specified rows in a tabular view, where *rows* is a sequence of structures, where each structure has this syntax:

```
structure
(logical-id: integer,
row-values: sequence,
background-color: background-color, {optional}
text-color: foreground-color {optional} )
```

gdu-tabular-view-sequence-of-structures-data-value-formatter
(*target*: item, *unformatted-values*: sequence,
dlg: gdu-dialog-instance, *control*: structure, *win*: g2-window)
–> *formatted-values*: sequence, *number-of-rows*: integer

Formats values stored as a sequence of structures. Each structure corresponds to a row, where the attributes in the structure correspond to a column name as specified in the *columns* argument. The sequence determines the order of the rows. For example:

```
sequence
    (structure
        (priority: 4,
        category: "Test",
        notes: "Test input sensor."),
    structure
        (priority: 2,
        category: "Report",
        notes: "Generate weekly metrics report."),
    structure
        (priority: 5,
        category: "Test",
        notes: "Check for viruses."))
```



Priority	Category	Notes
4	Test	Test input sensor.
2	Report	Generate weekly metrics report.
5	Test	Check for viruses.

gdu-tabular-view-sequence-of-sequences-data-value-formatter
(*target*: item, *unformatted-values*: sequence,
dlg: gdu-dialog-instance, *control*: structure, *win*: g2-window)
–> *formatted-values*: sequence, *number-of-rows*: integer

Formats values stored as a sequence of sequences. Each inner sequence corresponds to a row of values, where the number of values must correspond to the number of values in the *columns* sequence. The outer sequence determines the order of the rows. For example:

```
sequence
    (sequence ("2/1/2005", 20, 30, 50),
    sequence ("2/8/2005", 21, 32, 55),
    sequence ("2/15/2005", 24, 35, 60),
    sequence ("2/22/2005", 21, 29, 45),
    sequence ("2/29/2005", 24, 33, 51))
```

Date	Us	Ema	Asia
2/1/2005	20	30	50
2/8/2005	21	32	55
2/15/2005	24	35	60
2/22/2005	21	29	45
2/29/2005	24	33	51

gdu-tabular-view-sequence-data-value-formatter
(*target*: item, *unformatted-values*: sequence,
dlg: gdu-dialog-instance, *control*: structure, *win*: g2-window)
–> *formatted-values*: sequence, *number-of-rows*: integer

Formats values stored in a single sequence. The sequence describes the values, by row. For example, if N is the number of elements in the *columns* argument, then the values with an index 0 to N - 1 correspond with the first row, the values with an index N to 2 * N - 1 correspond with the second row, etc. For example:

```
sequence (3.4, 5.7, 77.0, 30.3, 3.0, 4.5, 80.0, 29, 7, 3.5, 5.5, 78.5)
```

Tag 1	Tag 2	Tag 3	Tag 4
3.4	5.7	77.0	30.3
3.0	4.5	80.0	29
7	3.5	5.5	78.5

gdu-tabular-view-structure-of-sequences-data-value-formatter
(*target*: item, *unformatted-values*: structure,
dlg: gdu-dialog-instance, *control*: structure, *win*: g2-window)
–> *formatted-values*: sequence, *number-of-rows*: integer

Formats values stored as a structure of sequences. Each sequence corresponds to a row of data within each column. The structure determines the data in each column, where the attributes in the structure correspond to a column name as specified in the *columns* argument. For example:

```
structure
    (date-time: sequence ("1/1/2005", "2/1/2005", "3/1/2005"),
    count-of-alarms: sequence (15, 35, 30),
    high-priority-alarms: sequence (3, 5, 4))
```

Date Time	Count Of Alarms	High Priority Alarms
1/1/2005	15	3
2/1/2005	35	5
3/1/2005	30	4

gdu-tabular-view-sequence-of-objects-formatter
(*target*: class item, *unformatted-values*: sequence,
dlg: class gdu-dialog-instance, *control*: structure, *win*: class g2-window)
–> *formatted-values*: sequence, *number-of-rows*: integer

Formats values stored as a sequence of objects, where each object in the sequence is a row in the tabular view. For example:

```
sequence
    (gdu-tabular-view-item-1,
    gdu-tabular-view-item-2,
    gdu-tabular-view-item-3)
```

Names	Prioirty	Message
GDU-TABULAR-VIEW-ITEM-1	12	First item
GDU-TABULAR-VIEW-ITEM-2	10	Second Item
GDU-TABULAR-VIEW-ITEM-3	4	Third Item

Text Box

gdu-add-text-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a text box control to the dialog, including a prompt label. Set *is-read-only* to true to make the text box a read-only display. For example, here is a text box that spans two vertical cells in the grid:

Here is a read-only text box:

See the Dynamic Dialog Specification example in `gdu-demo.kb`.

gdu-add-text-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog in a tab frame.

gdu-add-text-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-text-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *is-upper-case*: truth-value, *is-lower-case*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *format-value-procedure*: symbol, *unformat-value-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Set *is-upper-case* to true to force the text to upper case. Set *is-lower-case* to true to force the text to lower case. Specify *format-value-procedure* as a procedure to call to format the text and *unformat-value-procedure* as the procedure to call when the user presses the Backspace key. The syntax of the format and unformat value procedures is:

my-format-value-procedure
(*control*: structure, *val*: value)
–> *formatted-text*: text

Specifies *conclude-immediately* and *validation-procedure*.

gdu-add-text-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *is-upper-case*: truth-value, *is-lower-case*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *format-value-procedure*: symbol, *unformat-value-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog in a tab frame, and specifies *is-upper-case*, *is-lower-case*, *conclude-immediately*, *validation-procedure*, *format-value-procedure*, and *unformat-value-procedure*.

gdu-add-text-box-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *is-upper-case*: truth-value, *is-lower-case*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *format-value-procedure*: symbol, *unformat-value-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog with an anchor position when used in resizable dialogs. Also adds the control in a tab frame, and specifies *is-read-only*, *is-upper-case*, *is-lower-case*, *conclude-immediately*, *validation-procedure*, *format-value-procedure*, and *unformat-value-procedure*.

gdu-add-lower-case-text-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a text box control to the dialog that only accepts lower case text. Specifies *is-read-only*.

gdu-add-lower-case-text-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog that only accepts lower case, and displays it in a tab frame. Specifies *is-read-only*.

gdu-add-lower-case-text-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog that only accepts lower case, and displays it in a tab frame. Specifies *is-read-only*, *conclude-immediately*, and *validation-procedure*.

gdu-add-lower-case-text-box-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog that only accepts lower case text. Also specifies an anchor position when used in resizable dialogs, adds the control in a tab frame, and specifies *is-read-only*, *conclude-immediately*, and *validation-procedure*.

gdu-add-upper-case-text-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a text box control to the dialog that only accepts upper case text. Specifies *is-read-only*.

gdu-add-upper-case-text-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog that only accepts upper case, and displays it in a tab frame. Specifies *is-read-only*.

Following are the new API's added to involve text color and background color of the text box control in the GDU.kb :

1. gdu-add-text-box-control-with-colors (Dlg: class gdu-dialog-definition, control-id: value, attribute-name: symbol,text-color:symbol,background-color:symbol, initially-visible: truth-value, initially-enabled: truth-value, is-read-only: truth-value, grid-x: integer, grid-y: integer, grid-x-span: integer, grid-y-span: integer, tab-control: value, tab-name: text)

2. gdu-add-text-box-control-with-colors (Dlg: class gdu-dialog-definition, control-id: value, attribute-name: symbol,text-color:symbol,background-color:symbol, initially-visible: truth-value, initially-enabled: truth-value, is-read-only: truth-value, is-upper-case: truth-value, is-lower-case: truth-value, conclude-immediately: truth-value, validation-procedure: symbol, format-value-procedure: symbol, unformat-value-procedure: symbol, grid-x: integer, grid-y: integer, grid-x-span: integer, grid-y-span: integer)

3. gdu-add-text-box-control-with-colors(Dlg: class gdu-dialog-definition, control-id: value, attribute-name: symbol,text-color:symbol,background-color:symbol, initially-visible: truth-value, initially-enabled: truth-value, is-read-only: truth-value, grid-x: integer, grid-y: integer, grid-x-span: integer, grid-y-span: integer)

4. gdu-add-text-box-control-with-colors (Dlg: class gdu-dialog-definition, control-id: value, attribute-name: symbol,text-color:symbol,background-color:symbol, initially-visible: truth-value, initially-enabled: truth-value, is-read-only: truth-value, conclude-immediately: truth-value, validation-procedure: symbol, grid-x: integer, grid-y: integer, grid-x-span: integer, grid-y-span: integer, tab-control: value, tab-name: text)

5. gdu-add-text-box-control-with-colors (Dlg: class gdu-dialog-definition, control-id: value, attribute-name: symbol, text-color:symbol , background-color:symbol, initially-visible: truth-value, initially-enabled: truth-value, is-read-only: truth-value, is-upper-case: truth-value, is-lower-case: truth-value, conclude-immediately: truth-value, validation-procedure: symbol, format-value-procedure: symbol, unformat-value-procedure: symbol, anchor: value, grid-x: integer, grid-y: integer, grid-x-span: integer, grid-y-span: integer, tab-control: value, tab-name: text)

6. gdu-add-text-box-control-with-colors (Dlg: class gdu-dialog-definition, control-id: value, attribute-name: symbol,text-color:symbol,background-color:symbol, initially-visible: truth-value, initially-enabled: truth-value, is-read-only: truth-value, is-upper-case: truth-value, is-lower-case: truth-value, conclude-immediately: truth-value, validation-procedure: symbol, format-value-procedure: symbol, unformat-value-procedure: symbol, grid-x: integer, grid-y: integer, grid-x-span: integer, grid-y-span: integer, tab-control: value, tab-name: text)

Illustration of these new added api’s are present in GDU-DEMO . Open sub workspace Dynamic Dialog Specification on the top level work space. Click on the button Display Modal Dialog . You will see a dialog with text box controls with background colors and text colors too. The api calls are written in the following procedure on the same workspace.

```
_gdu-example-2-dialog-components(Target: class item {target item} , Dlg: class gdu-dialog-definition, Win: class g2-window)

{
}

begin

  { --- Build the list of components }
  call gdu-add-check-box-control (Dlg, the symbol activated, the symbol activated, true, true, 0, 0, 1, 1);
  call gdu-add-text-box-control-with-colors(Dlg, the symbol label, the symbol block-label,the symbol yellow , the symbol brown, true, true, false, 0, 1, 1, 2);
  call gdu-add-combo-box-control(dlg, the symbol mode, the symbol mode, true, true, false, sequence(), the symbol none, 0, 3, 1, 1);
  call gdu-add-spinner-control(Dlg, the symbol mean, the symbol mean, true, true, 0.0, 100.0, 1.0, 0, 4, 1, 1);
  call gdu-add-spinner-control(Dlg, the symbol variance, the symbol variance, true, true, 0.0, 100.0, 1.0, 0, 5, 1, 1);
  call gdu-add-duration-control (Dlg, the symbol sample-period, the symbol sample-period, true, true, 0, 6, 1, 1);

  call gdu-add-file-selection-control (Dlg, the symbol log-file, the symbol log-file, true, true, "csv", sequence(sequence("CSV Files (*.csv)", "*.csv"), sequence("All Files (*.*)", "*.*")), true, 0, 7, 1, 1);

  call gdu-add-list-box-control(dlg, the symbol queue, the symbol queue, true, true, false, false, sequence( "Messages", "warnings", "errors"), the symbol none, 0, 8, 1, 3);

  call gdu-add-text-box-control-with-colors (Dlg, the symbol status, the symbol status,the symbol red , the symbol black , true, true, true, 0, 10, 1, 1);
  call gdu-add-text-box-control(Dlg, the symbol counter, the symbol counter, true, true, true, 0, 11, 1, 1);

  call gdu-add-group-control(dlg, the symbol group-1, "Configuration", 0, 0, 1, 12);

  { --- The two metrics values to update dynamically }
  conclude that the gdu-control-ids-to-refresh of dlg = sequence(the symbol status, the symbol counter);

end
```

gdu-add-upper-case-text-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog that only accepts upper case, and displays it in a tab frame. Specifies *is-read-only*, *conclude-immediately*, and *validation-procedure*.

gdu-add-upper-case-text-box-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog that only accepts upper case text. Also specifies an anchor position when used in resizable dialogs, adds the control in a tab frame, and specifies *is-read-only*, *conclude-immediately*, and *validation-procedure*.

gdu-add-password-entry-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a text box control to the dialog that displays text as password characters. For example:

gdu-add-password-entry-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog that displays text as password characters and displays it in a tab frame.

gdu-add-password-entry-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a password text box control to a dialog, displays it in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-password-entry-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a password text box control to the dialog, specifies an anchor position when used in resizable dialogs, adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-symbol-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a text box control to the dialog that only allows the user to enter symbols, that is, no spaces. Specifies *is-read-only*.

gdu-add-symbol-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog that only allows the user to enter symbols, displays it in a tab frame, and specifies *is-read-only*. The validation procedure is automatically configured for this control to only allow symbols.

gdu-add-symbol-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *conclude-immediately*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog that only allows the user to enter symbols, displays it in a tab frame, and specifies *is-read-only* and *conclude-immediately*.

gdu-add-symbol-box-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *conclude-immediately*: truth-value, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control to the dialog that only allows the user to enter symbols, specifies an anchor position when used in resizable dialogs, displays it in a tab frame, and specifies *is-read-only* and *conclude-immediately*.

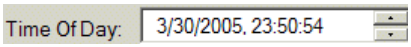
gdu-add-text-box-control-without-prompt
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *is-read-only*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a text box control without a prompt to the dialog. Also adds the control with an anchor position when used in a resizable dialog, adds the control in a tab frame, and specifies *is-read-only*, *conclude-immediately*, and *validation-procedure*.

Time of Day

gdu-add-time-of-day-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *show-time*: truth-value, *show-date*: truth-value, *use-24-hour-time*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a time of day control to the dialog, including a prompt label. Set *show-time* to true to show the time. Set *show-date* to true to show the date. Set *use-24-hour-time* to true to use a 24-hour clock. For example, here is a time of day control that shows the time and date, using a 24-hour clock:



See the Many Controls example in `gdu-demo.kb`.

gdu-add-time-of-day-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *show-time*: truth-value, *show-date*: truth-value, *use-24-hour-time*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a time of day control to a dialog in a tab frame.

gdu-add-time-of-day-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *show-time*: truth-value, *show-date*: truth-value, *use-24-hour-time*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a time of day control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-time-of-day-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *show-time*: truth-value, *show-date*: truth-value, *use-24-hour-time*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a time of day control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-read-only-timestamp-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a read-only text box control to the dialog, including a prompt label, which displays a timestamp.

gdu-add-read-only-timestamp-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a read-only timestamp control to the dialog in a tab frame.

gdu-add-read-only-timestamp-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a read-only timestamp control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

Toggle Button

gdu-add-toggle-button-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *label*: text, *icon-name*: item-or-value, *use-cell-width*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a toggle button to the dialog. Specify *label* as the text to display in the push button, and specify *icon-name* as the class name whose icon description to use as an icon in the toggle button instead of the label. Set *use-cell-width* to true to use the full width of the grid cell for the control, which left-aligns the control in the cell.

Specify *activation-procedure* as the procedure to call when the user clicks the toggle button. The activation procedure must have this syntax:

activation-procedure
(*dlg*: class gdu-dialog-instance, *control*: structure, *other-values*: sequence, *target*: class item, *win*: class g2-window)

gdu-add-toggle-button-control
(*dlg*: class gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *label*: text, *icon-name*: item-or-value, *use-cell-width*: truth-value, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a toggle button control to the dialog in a tab frame.

Track Bar

gdu-add-track-bar-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a track bar control to the dialog, including a prompt label. A track bar is a slider with tick marks. Specify *min-value*, *max-value*, and *increment* as the minimum and maximum values and the amount to increment. For example:

See the Many Controls example in `gdu-demo.kb`.

gdu-add-track-bar-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a track bar control to the dialog in a tab frame.

gdu-add-track-bar-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a track bar control to the dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-track-bar-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *min-value*: integer, *max-value*: integer, *increment*: integer, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a track bar control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

Tree View Combo Box

gdu-add-tree-view-combo-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *tree-structure*: structure, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a tree view combo box control to the dialog, including a prompt label. Specify *tree-structure* as a structure that specifies the tree. Only one item can be selected at a time in the combo box, which can be any node in the tree. The advantage of using a tree view compared to a flat list is that it enables you to

organize the list of choices of the control in a hierarchical format. The *tree-structure* has this syntax:

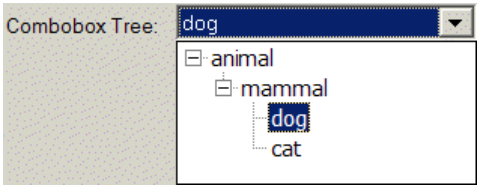
(item-or-name: *item-or-text*,
children: *sequence-of-structures*)

where:

- *item-or-text* is any named G2 item or any text string.
- *sequence-of-structures* is a sequence of structures with this format:

```
sequence
  (structure
    (item-or-name: item-or-text,
    children: sequence-of-structures)
   . . . )
```

The top-level item-or-name in the structure is the parent node in the tree view, and the top-level children are the children of the parent node. The tree structure can be nested as many levels deep as needed to describe the tree. For example:



See the Many Controls example in `gdu-demo.kb`.

gdu-add-tree-view-combo-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *tree-structure*: structure, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a tree view combo box control to a dialog in a tab frame.

gdu-add-tree-view-combo-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *tree-structure*: structure, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a tree view combo box control to a dialog in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

gdu-add-tree-view-combo-box-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *tree-structure*: structure, *conclude-immediately*: truth-value, *validation-procedure*: symbol, *grid-x*: integer, *grid-y*: integer, *anchor*: value, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a tree view combo box control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

Workspace

gdu-add-workspace-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *workspace-name-or-uuid*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer)

Adds a workspace view control to the dialog. If *use-cell-width* is set to true, no prompt is displayed and the workspace view will cover the full cell area and other cells if the x or y span is greater than 1. *workspace-name-or-uuid* can be a symbol to specify the name of the workspace or a text containing the uuid of the workspace. For example:

See the Workspace example in `gdu-demo.kb`.

gdu-add-workspace-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *workspace-name-or-uuid*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a workspace view control to a dialog in a tab frame.

gdu-add-workspace-control
(*dlg*: gdu-dialog-definition, *control-id*: value, *attribute-name*: symbol, *initially-visible*: truth-value, *initially-enabled*: truth-value, *use-cell-width*: truth-value, *workspace-name-or-uuid*: value, *anchor*: value, *grid-x*: integer, *grid-y*: integer, *grid-x-span*: integer, *grid-y-span*: integer, *tab-control*: value, *tab-name*: text)

Adds a workspace control to the dialog with an anchor position when used in a resizable dialog. Also adds the control in a tab frame, and specifies *conclude-immediately* and *validation-procedure*.

4

Converting GUIDE Dialogs

Describes how to convert GUIDE dialogs to GDU dialog definitions.

Introduction

The G2 Dialog Conversion Utility (GDUC) provides automatic conversion of GUIDE/UIL dialogs to Windows dialog definitions for viewing in Telewindows.

Note The dialog conversion tool provides a starting point for migrating GUIDE/UIL dialogs to standard Windows dialogs. Some manual configuration will be required to convert all features of GUIDE dialogs, but these tools should ease the migration.

GDUC creates **dialog definitions**, which are instances of the gdu-dialog-definition class, from uil-dialog objects. Once you have converted the dialogs and edited the specification to fine-tune the conversion, you launch these dialogs by calling a procedure, which creates a **dialog instance**, an instance of the gdu-dialog-instance class.

GDUC manages the converted dialog for you. However, you can also customize various methods for the overall dialog, as well as for individual controls. For details, see the following sections in this chapter.

Converting GUIDE/UIL Dialogs Interactively

To convert GUIDE/UIL dialogs interactively:

1 Load this file:

Windows	`g2i\kbs\gduc.kb`
UNIX	`g2i/kbs/gduc.kb`

2 Merge into this application the KB whose dialogs you want to convert.

3 Navigate to the uil-dialog object you want to convert.

4 In uil-build mode, choose convert uil dialog to native dialog on the dialog object.

A gdu-dialog-definition object appears on top of the UIL dialog object, slightly offset.

5 Choose display converted dialog on the gdu-dialog-definition object to display the dialog in the client.

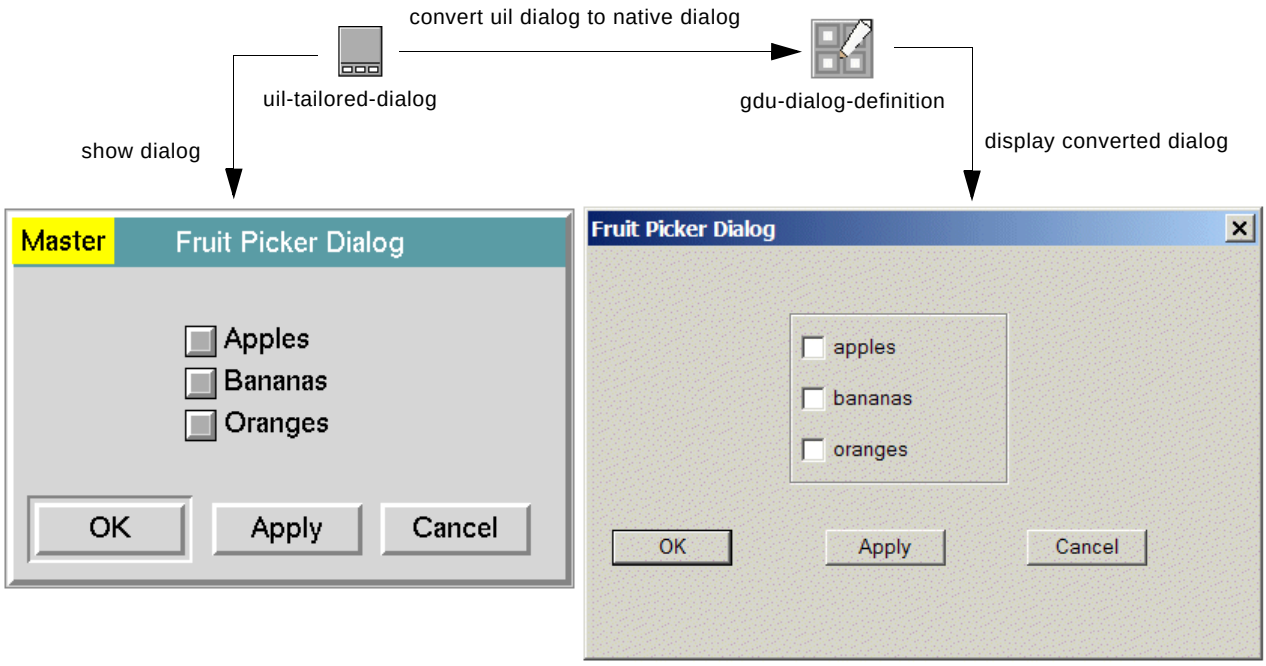
6 Check the dialog layout for accuracy.

The size and/or position of the dialog controls might need to be edited.

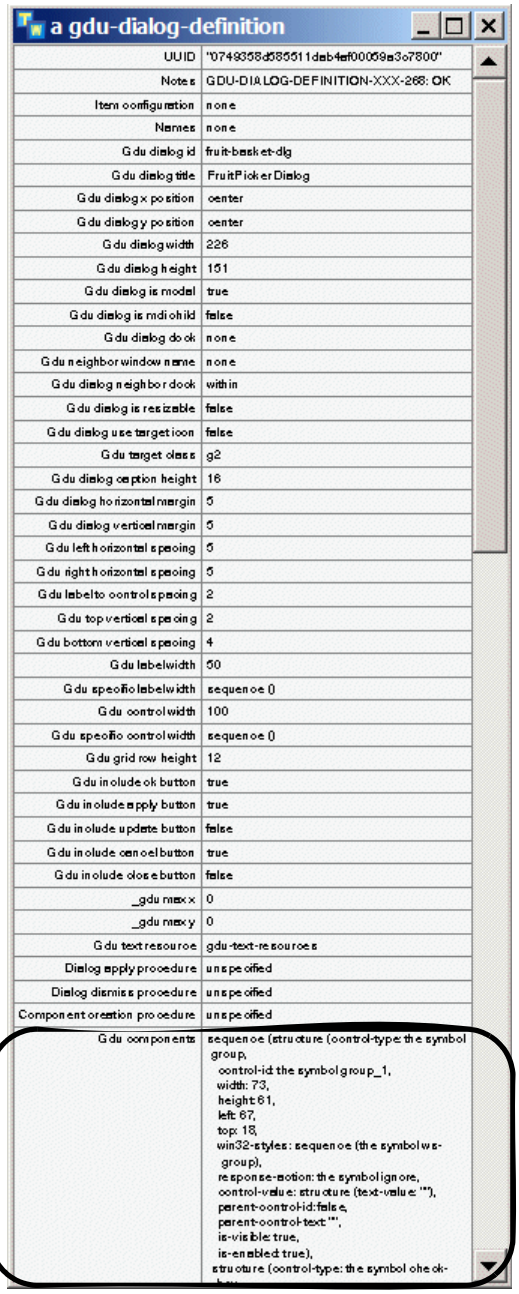
7 Edit the dialog specification, as needed.

You can edit the dialog specification manually by editing the gdu-components attribute of the gdu-dialog-definition. You can also use the dialog configuration editor to edit the specification interactively by choosing properties on the gdu-dialog-definition. See Chapter 5, “Dialog Definition Editor” on page 115.

For example, here is the master dialog for a uil-tailored-dialog and its converted gdu-dialog-definition:



Here is the attribute table for the converted gdu-dialog, which provides the dialog specification. It includes the gdu-components attribute, which is visible in administrator mode.



Converting GUIDE/UIL Dialogs Programmatically

To convert GUIDE/UIL dialogs programmatically:

➔ gduc-convert-uil-dialog
(*dialog*: class uil-tailored-dialog, *win*: class g2-window)
–> *dialog*: class gdu-dialog-definition

Converts a uil-tailored-dialog to a gdu-dialog-definition.

5

Dialog Definition Editor

Describes how to use the dialog definition editor.

Introduction

The G2 Dialog Configuration Editor (GDUE) module provides a Windows editor for editing the properties of a dialog definition object. The editor allows you to edit the specification of individual controls, as well as the overall dialog.

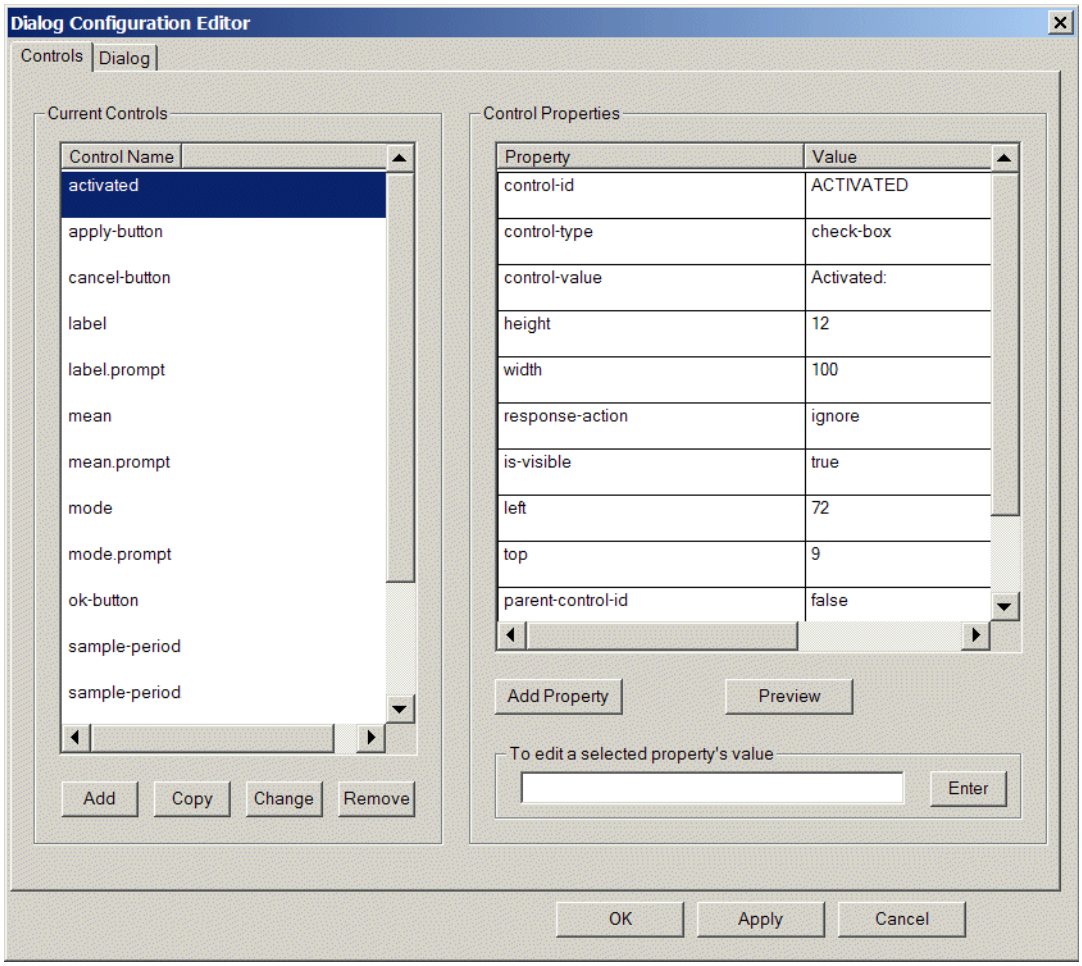
To launch the Dialog Configuration Editor:

1 Merge this file into the application that contains gdu-dialog-definition instances that you want to edit:

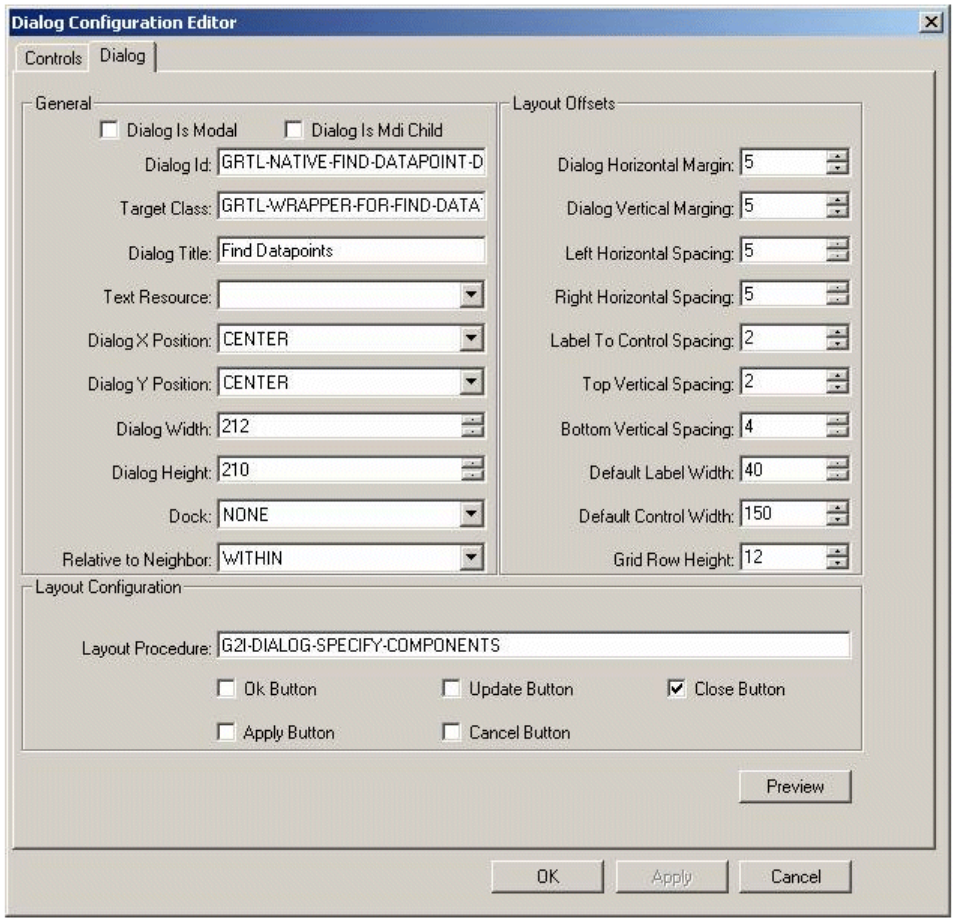
Windows	`g2i\kbs\gdue.kb`
UNIX	`g2i/kbs/gdue.kb`

2 Choose properties on a gdu-dialog-definition object.

Here is Controls tab of the dialog configuration editor for editing individual controls in the gdu-components attribute of a dialog definition object:



Here is Dialog tab of the dialog configuration editor for editing properties of the overall dialog definition:



Index

I

L

M

P

R

S

T

W