G2 Dynamic Displays

User's Guide

Version 2015

G2 Dynamic Displays User's Guide, Version 2015

December 2015

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: DOC020-1200

Contents

Preface

Describes this document and the conventions that it uses.

About this Guide

This guide contains complete information about the G2 Dynamic Displays Utility (GDD) and shows you how to use GDD at any supported level. GDD provides a library of animated icons that you can easily customize and embed into other G2 objects. You can use animated meters, dials, and gauges to display real-time application information. The GDD engine, which incorporates four runtime procedures in a module called GDDROOT, is used for deployment. The GDD Library and GDD Development Tools allow you to interactively customize animated icons as well as to design your own dynamic display icons from scratch.

This guide:

- Introduces G2 Dynamic Displays and describes the capabilities that it provides.
- Describes the GDD user interface and shows you how to use it to define customized displays graphically.

- Lists all GDD API functions and their signatures in a reference dictionary.
- Includes a glossary of all GDD terms and concepts.

This guide contains specific instructions for using GDD. These instructions will not work unless you have loaded GDD, as described under Installing GDD.

Audience

This guide assumes that you are generally familiar with G2 terminology and practices but does not require a thorough understanding of G2. If you encounter G2 terms or concepts that you do not understand, see the *G2 Reference Manual*.

Organization

This guide contains five chapters:

	Title	Description
1	Overview of G2 Dynamic Displays	Summarizes the features of the G2 Dynamic Displays utility and describes the ways you can use GDD to add customized dynamic displays to your application.
2	Getting Started	Describes the requirements for running GDD, how to install GDD, the module structure of GDD, and the GDD Demo.
3	Creating a Customized Attribute Display	Shows how to use GDDDEV to build a customized dynamic display.
4	GDD Class Information Objects	Describes each attribute and its value for the three kinds of class information objects
5	The GDD API Reference	Describes all supported GDD API calls.

A Note about the API

The GDD API, as described in this user's guide, is not expected to change significantly in future releases, but exceptions may occur. A detailed description of any changes will accompany the GDD release that includes them.

Therefore, it is essential that you use GDD exclusively through its API, as described in this user's guide. If you bypass the API, you cannot rely on your code to work in the future, since GDD may change, or in the present, because the code may not correctly manage the internal operations of GDD.

If GDD does not seem to provide the capabilities that you need, contact Gensym Customer Support at 1-781-265-7301 (Americas) or +31-71-5682622 (EMEA) for further information.

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

Convention Examples	Description
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
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 Users' 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 OPCLink User's Guide*
- *G2 PI Bridge User's Guide*
- *G2-SNMP Bridge User's Guide*
- *G2 CORBALink 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

Overview of G2 Dynamic Displays

Summarizes the features of the G2 Dynamic Displays utility and describes the ways you can use GDD to add customized dynamic displays to your application.

Introduction

This chapter introduces **G2 Dynamic Displays Utility** (**GDD**) and defines GDD terms and concepts. Be sure you are familiar with the information in this chapter before you read the rest of this guide.

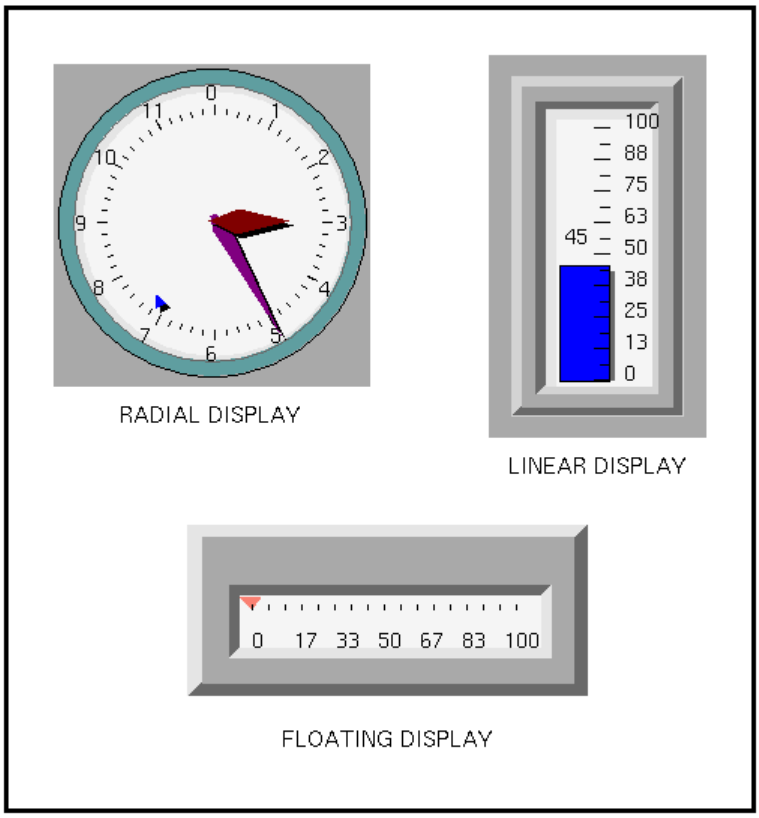
What is GDD?

GDD is a G2 Utility that gives you the ability to include dynamically generated graphical attribute readouts, called **dynamic displays**, to icons in your applications.

You can include three basic kinds of display in your application:

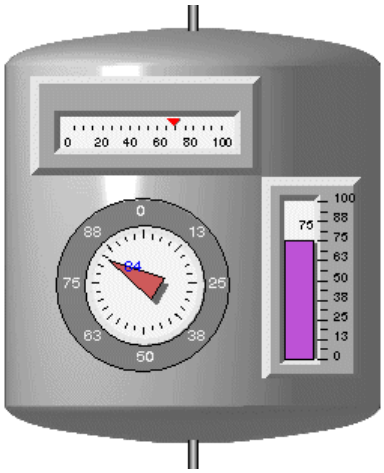
- Radial display
- Linear display
- Floating display

You can use these displays to incorporate behavior consistent with meters, gauges, and dials in your existing application, or include them in an application under development. GDD enables you to customize animated regions in the icon of any arbitrary class definition. The following figure shows a sample of each kind of display.



An icon can have multiple graphical readouts. For example, a single tank might have one display of each type. It might display temperature on a floating display, pressure on a radial display, and volume on a linear display. Each display has a different range of measurement and can have a label identifying the property being displayed.

The following figure shows a tank with three different dynamic displays:



You can use GDD three ways:

- Design a customized dynamic display for your application.
- Customize a meter, gauge, or dial from GDD's library of default dynamic displays.
- Use API calls to manage and control dynamic displays.

Components of GDD

GDD is a knowledge base (KB) module, consisting of three KBs, each having its own top-level workspace. The three components of GDD are:

- G2 Dynamic Displays Development (gdddev)
- G2 Dynamic Displays Library (gddlib)
- G2 Dynamic Displays Root (gddroot)

All components of GDD are identified by either the public gdd- prefix or private _gdd- prefix.

Caution Users should never directly call any procedure prefixed by an underscore. These procedures are part of the implementation and could change without warning.

The Development Component

The Development component of GDD is called **GDD Development**. It is in a module called gdddev. Use gdddev when you are developing dynamic displays. This module contains the tools for developing new GDD displays. Once the

dynamic display has been created, the gdddev module can be removed from your KB.

The gdddev top-level workspace contains:

- Access to a list of the **Application Programmers Interface(API)** procedures that are used in developing dynamic displays.
- A palette of GDD **Class Information Objects** that contain in their tables information about the properties of a dynamic display. These properties include:
  - The class of object whose attributes are displayed in the dynamic display.
  - The attribute of that object to display.
  - The range of values on the display.
  - The number of and length of the tick marks on the display.
  - The labels on the display.
  - The appearance and movement direction of the animated part of the display.

The Library Component

The library component of GDD is called the **GDD Library**. It is in a module called gddlib. Use gddlib when you are developing dynamic displays and wish to use standard displays from the GDD library of default meters in your application with only a small amount of customization. The GDD library contains samples of the three basic kinds of displays. Each kind of display can easily be customized by using the icon editor and updating the attributes.

The displays in the GDD library use several of the icon features introduced in G2 5.0. These features include animating icons and adding a background layer to the icon. For additional information about G2 icons, see the *G2 Reference Manual*.

Each object class definition in the library has a subworkspace containing a gdd-class-information-object which stores all of the class specific information for the data display.

If the icon of an item contains several data displays, GDD uses a different class-information-object for each display. All of the class information objects are stored in a subtable of the object definition. In this way all of the information about the data display is stored in the class definition.

Runtime Component

The runtime component of GDD is called **GDD Root**. It is in a module called gddroot. Use gddroot when you are running applications that include dynamic displays. The top-level workspace of gddroot provides access to a list of the

signatures of the API procedures necessary to manage dynamic displays after they are developed.

Internationalizing a Dynamic Display

If you want your display text to appear in a language other than English, you can use Gensym Foundation Resources (GFR) to internationalize GDD displays. For information on using GFR to internationalize your KB, see the *G2 Foundation Resources User's Guide*.

GDD Online Documentation

Using the G2 OnLine Documentation (GOLD) utility, you can view GDD documentation on a Web browser from G2. A separate module, GOLDUI, provides the user interface for setting up and using GOLD. See the *G2 OnLine Documentation User's Guide* for instructions on installing and setting up GOLD for online viewing of GDD documentation.

2

Getting Started

Describes the requirements for running GDD, how to install GDD, the module structure of GDD, and the GDD Demo.

Introduction

This chapter provides an overview of how to install GDD, access its features, run the demos, and access online help.

Installing GDD

You install GDD by merging any of its KBs into any modularized knowledge base. When you merge a GDD KB, its required modules are automatically loaded into G2.

The file names of the GDD KBs are:

- *gdddev.kb*
- *gddlib.kb*
- *gddroot.kb*

The default location of these KBs is the *utils* subdirectory in the *kbs* directory under the *g2* directory.

To merge a GDD KB into your KB:

1. Pause or reset your KB.
2. Choose Merge KB from the Main Menu to display the Load KB workspace.

   The merge in this KB option is selected.
3. Specify the location of the GDD KB file you wish to use and click End.

Tip When merging a GDD KB, let G2 resolve conflicts by enabling the automatically resolve conflicts option.

When you merge a GDD KB into your KB, it is not a required module unless it is specified in the Module Information table of your KB.

To make GDD a required module:

1. Choose Main Menu > System Tables > Module Information.
2. Specify the GDD KB as a directly required module of the appropriate existing module of your KB.

You would most often do this for gddroot since it is required to run applications which use dynamic displays. For more information on merging KBs and making a KB a required module, see the *G2 Reference Manual*.

GDD Required Modules

When you merge one of the GDD KBs into your KB, its required modules are automatically loaded into G2. The following table describes these modules:

Module	File Name	Required Modules	Contents
gdddev	*gdddev.kb*	*gddroot* *gfr* *sys-mod* *uilroot*	Contains the development tools needed to create your own dynamic displays.
gddroot	*gddroot.kb*	*gfr* *sys-mod* *uilroot*	Contains the procedures, classes and methods necessary to run GDD displays.
gddlib	*gddlib.kb*	*gddroot* *gfr* *sys-mod* *uilroot*	Contains a library of working dynamic displays for use in any G2 application.
gfr	*gfr.kb*	*sys-mod* *uilroot*	Definitions and API support for the G2 Foundation Resources (GFR) utility.
sys-mod	*sys-mod.kb*	*uilroot*	The library of G2 system procedures.
uilroot	*uilroot.kb*	none	Definitions and API support for navigation buttons.

Note These module dependencies are subject to change in future versions of GDD.

Starting GDD

GDD works only when G2 is running. After merging a GDD KB, resume or start G2.

To start GDD:

➔ Choose Resume or Start from the G2 Main Menu.

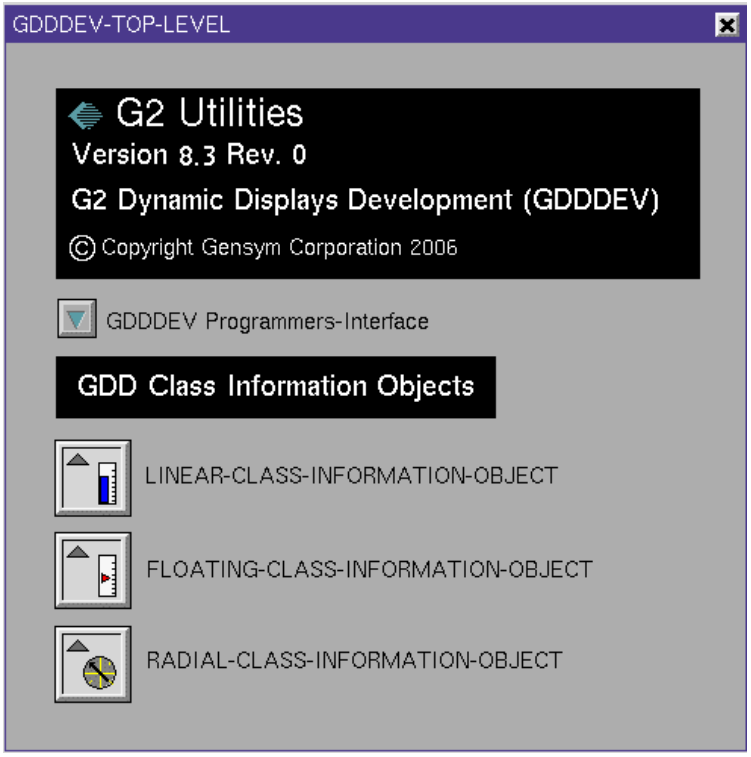
Accessing GDD Features

After merging one of the GDD KBs and starting G2, you can access the GDD features from the top-level workspace of that KB. From the gdddev top-level workspace you can:

- Access a list of procedure signatures for the part of the Application Programmer's Interface (API) used to develop dynamic displays.
- Build custom dynamic displays using a palette of GDD class information objects.

To display the gdddev top-level workspace:

➔ Choose Main Menu > Get Workspace > gdddev-top-level.



Clicking on the navigation button labeled GDDDEV Programmer's Interface displays a workspace containing procedure signatures for those procedures useful in developing new dynamic displays. The class information objects can be cloned to use in your own application.

From the gddlib top-level workspace you can select the type of component you wish to clone from the GDD library.

To display the gddlib top-level workspace:

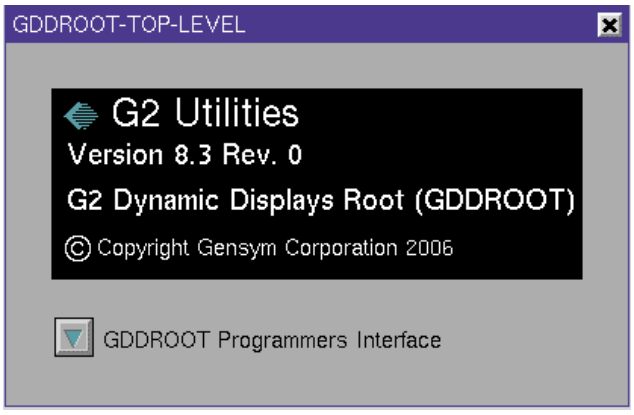
➔ Choose Main Menu > Get Workspace > gddlib-top-level.

Clicking on one of the navigation buttons displays a workspace containing standard dynamic displays of the type indicated. These displays can be cloned and modified for your application.

From the gddroot top-level workspace you can access the part of the Application Programmer's Interface used to manage dynamic displays.

To display the gddroot top-level workspace:

➔ Choose Main Menu > Get Workspace > gddroot-top-level.



Clicking on the navigation button labeled GDDROOT Programmer's Interface displays a workspace containing procedure signatures of procedures useful in managing dynamic displays.

For information on...	See...
Building Custom Displays	Creating a Customized Attribute Display.
The Application Programmer's Interface	The GDD API Reference.

The GDD Demo KB

GDD Demo is a KB of examples that cover the main features of GDD. This KB, named gdddemo, is located in the *utils* subdirectory of the *kbs* directory under the *g2* directory.

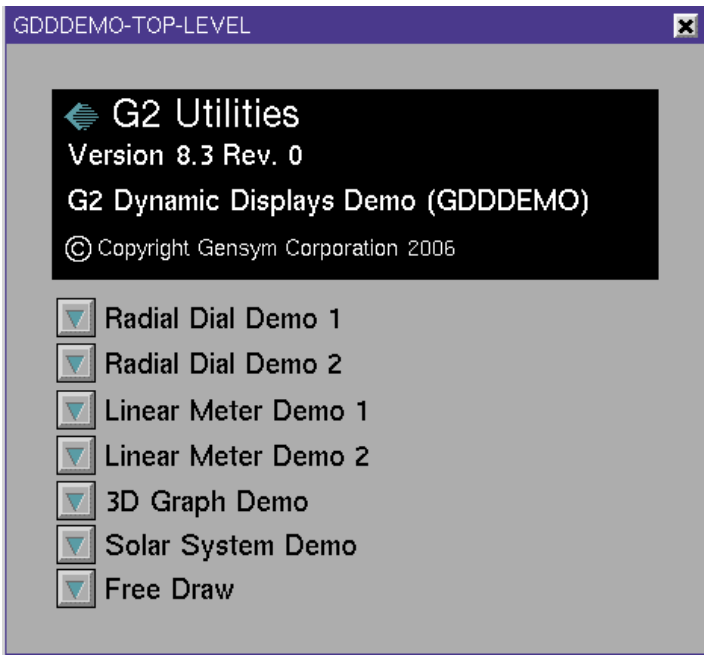
To view the GDD examples:

1 Load the *gdddemo.kb* or merge it into your KB.

Like the other GDD KBs, the demo KB loads all of the modules it requires. The required modules for gdddemo are gdddev and gddlib, which both require gddroot.

➔ Choose Main Menu > Get Workspace > gdddemo-top-level.

After resuming G2, the GDD Demo workspace appears:



This workspace contains the following subworkspaces:

Title	Description
Radial Dial	Example showing how to create a radial dynamic display.
Radial Dial 2	Example showing how to create a variation on a radial display.
Floating Meter	Example showing how to create a floating display.
Linear Meter	Example showing how to create a linear display.
3D Graph	Example showing how to create a three dimensional display.
Solar System	Example showing the use of the GDD API procedure gdd-rotate-icon.
Free Draw	Example showing the use of the GDD API procedures gdd-activate-icon, gdd-rotate-icon, and gdd-shift-icon.

As you browse the demo KB, you can see how to construct GDD displays and how to define certain behaviors by setting attributes and using API calls.

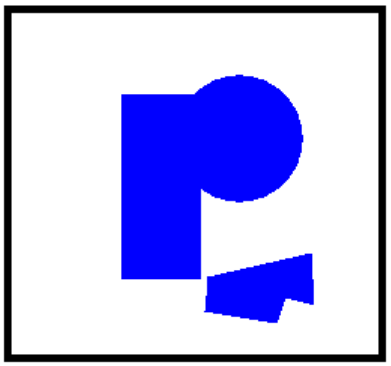
The Free Draw Demo needs some explanation to use. It is a demonstration of how to use the GDD API to activate and move the active regions of any icon. You need to create the active regions of the icon before you can activate and move them.

To use the Free Draw Demo:

1. Click on the navigation button to display the Free Draw workspace.
2. Choose edit icon from the menu of the object definition of the Generic class.
3. From the icon editor, add a new region to the icon and give the region a name.

4 Use the drawing buttons to draw several shapes and place them on the new layer.

For example:



5 Set the color of the new region so that the shapes display in a bright color.

6 Click the End button to dismiss the icon editor.

7 On the Free Draw workspace, click the Activate button.

The Activate button calls gdd-make-icon-active. This procedure activates every named region in the icon.

8 Experiment with the group of buttons on the workspace by moving and rotating the icon.

9 Once you have observed the effect of each button, pause G2 and display the table for each button to see the procedure call used in each case.

10 Add another named region to the icon, using a different arrangement of shapes and a different color.

11 Repeat steps 7 and 8.

To close the GDD demo workspace:

➔ Choose Hide Workspace from the workspace menu.

To redisplay the GDD demo workspace:

➔ Choose Main Menu > Get Workspace > gdddemo.

Removing the Demo Module

Because the GDD module of examples is not a required module, you may delete it when you finish viewing the GDD examples.

To delete the gdddemo module:

1. Choose Main Menu > Miscellany > Delete Module to display a menu of the modules currently loaded in G2.
2. Choose gdddemo to display a confirmation dialog.
3. Click All to delete the module and all its associated workspaces.

Accessing the GDD Online Documentation

Once you have installed the G2 OnLine Documentation (GOLD) utility, you can access GDD documentation from G2 and view it on a Web browser.

GOLD provides online viewing of HTML (HyperText Markup Language) files, which correspond to the printed documentation in the *G2 Dynamic Displays User's Guide*. HTML files are standard files, which any HTML browser on any platform can display.

When G2 is running, you can access the online documentation associated with GDD in the following ways:

- Choose Help from the G2 Main Menu.
- Position your cursor on a GDD item and press the F1 key for context-sensitive help.

Note For context-sensitive help to work correctly, the user mode can be any mode *except* administrator.

3

Creating a Customized Attribute Display

Shows how to use GDDDEV to build a customized dynamic display.

Introduction

This chapter shows how to build a dynamic display, using the GDD Development Module, gdddev. The process has several steps:

- Create an object definition.
- Create an icon for the new class of object.
- Create an instance of the class.
- Clone a gdd-class-information-object from the gdddev palette.
- Edit the attributes of the gdd class information object.

- Use gdd-setup to insert the information about the display from the table of the gdd class information object into the class definition.
- Create a procedure or rule to drive the display.
- Activate the display.

To see online examples that illustrate the ideas in this chapter:

1 Load the file *gdddemo.kb* from the *kbs* directory.

2 Choose the following demos:

- Radial Dial
- Radial Dial 2
- Floating Meter
- Linear Meter

The principles described in this chapter apply to all GDD displays. For simplicity, the chapter focuses on radial displays. Creating Linear and Floating Displays describes special techniques for linear and floating displays.

Creating an Object Definition

Create a new object definition and give it the necessary attributes. The figure below shows the object definition for the object that will be used throughout this example.

MY-DIAL

To create a new object definition:

1 Choose KB Workspace > New Definition > class-definition > object-definition, and place the new object on your definitions workspace.

2 Provide values for the following attributes of the new object:

- class-name
- direct-superior-classes
- class-specific-attributes

The new class can inherit from an existing user-defined class or from object.

Configure class-specific-attributes only if you wish to define a new attribute whose value the dial displays. If you wish to display one of the object's inherited attributes, you do not need to define any class specific-attributes.

Creating an Icon for the Class

Next, you create an icon for the class.

To create an icon for the class:

➔ Choose edit icon from the menu of the object definition.

This displays the icon editor. Use the icon editor to create an icon for the object. The icon should include active as well as inactive regions.

One of the features of icons is **icon variables**. Icon variables are user-defined variables that specify almost any element of a class's icon description. Using the conclude action, you can change the value of an icon variable in an instance of the class. The instance's icon immediately changes to reflect the new value.

A common use of icon variables is to specify the positions of the defining points of a graphical icon component, for example, the pointer in a radial dynamic display. You can then cause the icon component to move by changing the values of the icon variables. A region of an icon whose coordinates are defined by icon variables is called an **active region**.

The active regions are in those layers that define the pointer for the dial, or other moving part of the display. Be sure to name all of the layers that will be active regions. GDD uses a sequence of the names of these layers to obtain the regions of the icon to activate, and then later, to move to an appropriate value when the icon is updated.

Caution When naming the icon layers for the active regions of the display, make sure that no name is a substring of any other name. For example, pointer-1 and pointer-2 are fine because they differ in the last character. The names point and pointer are not acceptable because the shorter name point "shadows" the longer name pointer.

For more information about using the icon editor, refer to the chapter on icons in the *G2 Reference Manual*.

Instead of creating your own icon, there are times you might like to copy an existing dynamic display.

To copy the display from another object definition:

➔ Copy the value of the icon-description attribute from the table for the object whose icon you wish to copy and paste the copied value into the icon-description attribute of your new object.

or

➔ Find a dynamic display in the GDD Library and clone its object definition into your workspace.

Creating an Instance of the Class

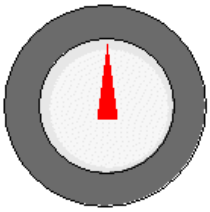
To see the new icon, you create an instance of the class.

To create an instance of the class:

1 Choose create instance from the table of the object definition.

2 Paste the new icon on your definitions workspace.

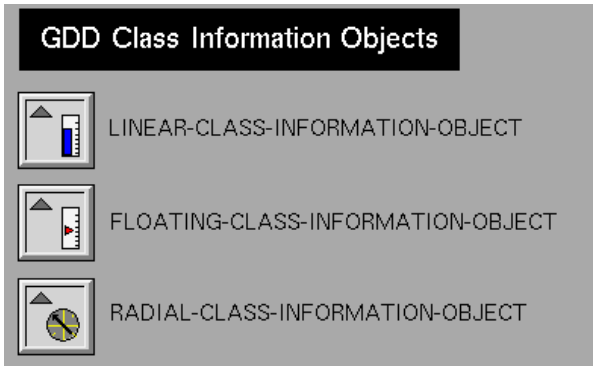
The figure below shows the icon for my-dial.



Setting the Attributes of the New Dynamic Display

The first step in setting the attributes of the dynamic display is to clone the appropriate GDD information object and place it in your definitions workspace near the class definition.

Display the gdddev-top-level workspace. This workspace provides the tools for developing your own dynamic displays. At the bottom of the workspace is a **palette** containing the three kinds of class information objects, one for each type of dynamic display. You clone the appropriate information object from the palette and use its table to set the properties of your display. The figure below shows the gdddev palette:



Cloning Information Objects from the Palette

To clone information objects from the palette to a workspace:

1. Display the gdddev-top-level workspace.
2. Click the mouse button over the information object.

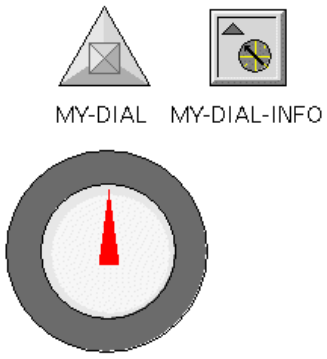
   A copy of the information object appears, attached to the mouse pointer.
3. Move the mouse cursor to the desired location on a workspace.

Note A convenient place to locate the class information object is in a subworkspace of the object definition, however, you can locate it on any convenient workspace.

4. Click the mouse button to place the information object on the workspace.

   The information object transfers to the workspace at the location you choose.

The following figure shows the workspace as it looks at this point:



The next set of instructions shows you the attributes to set on the class information object. For information on the effect of each attribute and its allowable and default values, see the tables in GDD Class Information Objects.

Associating Information Objects with Dynamic Displays

To associate the information object with your dynamic display:

1. Display the attribute table of the information object.
2. Edit the gdd-target-class attribute to specify the name of the class you have created.
3. Edit gdd-target-attribute to specify the attribute of the class you wish to associate with this display.

4 If you wish, you may name the information object by assigning a value to the names attribute.

If a name for the information object is not provided by the user, then, during setup, GDD assigns a name, based on the class name provided to the gdd-target-class attribute:

Names	MY-DIAL-INFO
Gdd status	none
Gdd target class	my-dial
Gdd target attribute	level

Setting up the Center of the Display

To set up the horizontal and vertical center of the new display:

➔ Assign values to the gdd-horizontal-center attribute and the gdd-vertical-center attribute.

This will specify the x and y location of the center of the dial. The values are given in workspace units, for example:

Gdd horizontal center	100
Gdd vertical center	100

Setting the Minimum and Maximum Values

To set up the minimum and maximum value for the dial:

➔ Assign values to the gdd-minimum-value attribute and the gdd-maximum-value attribute.

This will specify the range of numbers displayed on the dial:

Gdd minimum value	0
Gdd maximum value	100

Setting up Tick Marks

To set up the tick marks on the display:

1 Edit the gdd-tickmark-count attribute to specify the number of tick marks that appear in the display.

The **tick marks** represent intermediate positions between the maximum value and the minimum value on the display.

2 Edit the gdd-tickmark-color attribute.

The tick marks can be any G2 color.

3 Edit the gdd-tickmark-length attribute.

The length of the tick mark is given in workspace units.

4 Edit the gdd-major-tickmark-count attribute to specify the number of major tick marks that appear in the display.

This attribute designates how many of the tick marks are major tick marks. The major tick marks are evenly distributed throughout the range of tick marks. Major tick marks are often labeled and represent just some of the values in the range.

5 Edit the gdd-major-tickmark-length attribute.

Major tick marks are generally longer than minor tick marks.

Gdd tickmark count	16
Gdd tickmark color	black
Gdd tickmark length	4
Gdd major tickmark count	4
Gdd major tickmark length	10

Next, set up the properties of the labels that associate a value with the major tick marks.

Setting up Labels

To set up the labels on the display:

1 Edit the gdd-label-count attribute to specify number of labels that appear in the display.

2 Edit the gdd-label-color attribute.

The label can be any G2 color.

3 Edit the gdd-label-size attribute to specify the size of the font used in the label.

The choices of font size are small, large, and extra large.

4 Edit the gdd-label-offset attribute to specify the distance between a label and the base of its tickmark.

Gdd label count	8
Gdd label color	smoke
Gdd label size	small
Gdd label offset	20

This example includes a tracking label in the display. A **tracking label** is an optional feature that displays the current value of the target attribute.

Including Tracking Labels

To include a tracking label in the display:

1 Edit the gdd-tracking-label attribute to specify if the display has a tracking label.

A value of true will include a tracking label in the display, false excludes the label.

2 Edit the gdd-tracking-label-color attribute to specify the color of the label.

The label can be any G2 color.

3 Edit the gdd-tracking-label-size attribute to specify the size of the font used in the tracking label.

The choices of font size are small, large, and extra large.

4 Edit the gdd-tracking-label-offset attribute to specify the distance between the center of the radial display and the tracking label in workspace units.

Gdd tracking label	true
Gdd tracking label color	blue
Gdd tracking label size	small
Gdd tracking label offset	30

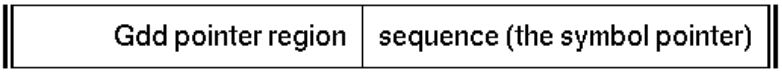
The next step is to indicate the active regions in the icon. When drawing the icon, you decide which layers of the icon will be animated. These are the layers that define the pointer for the dial. The gdd-pointer-region attribute of the class information object is a sequence that you edit to contain the names of these layers. This sequence is used by the GDD routines to obtain the layers of the icon to activate, and then later, to rotate the pointer to an appropriate value when gdd-update is called.

Indicating Active Regions

To indicate the active regions in the icon:

➔ Edit the gdd-pointer-region attribute to specify the names of the active regions in the display.

As shown in the following figure, the value is a sequence containing the names of all of the active icon regions.



For additional information about using sequences see the *G2 Reference Manual*. These regions are associated with the pointer in radial and floating displays.

The remaining attributes are specific to radial displays. Radial displays are either circles or circular arcs, which are segments of circles, measured in degrees. A half-circle is an arc of 180 degrees; a full-circle is an arc comprising 360 degrees.

Setting up Radial Information

To set up the radial information:

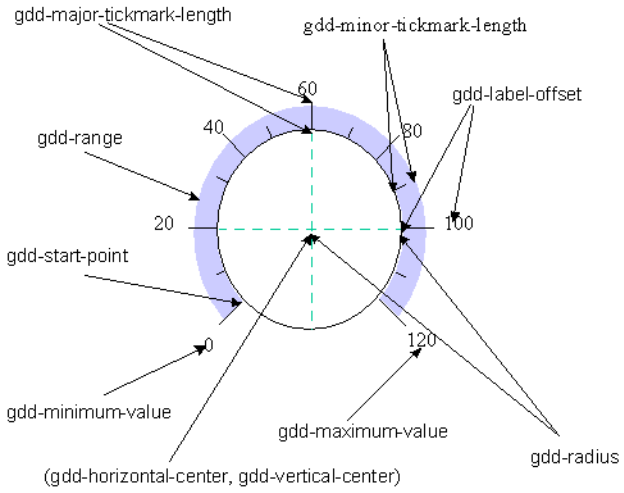
1. Edit the gdd-radius attribute to specify the radius of the display in workspace units.
2. Edit the gdd-range attribute to specify the shape of the display.

   This variable uses degrees of a circle to describe the shape of the display. An acceptable value is any quantity between 0.0 and 360.0.
3. Edit the gdd-start-point attribute to specify the starting location of the pointer in terms of the attribute value.

   An allowable value for this attribute is any value between the gdd-minimum-value and the gdd-maximum-value.
4. Give the gdd-clockwise attribute the value true to specify that the pointer moves clockwise.

   False indicates that the pointer moves counterclockwise.

The following figure summarizes the attributes of a **radial class information object** and the features of a radial dynamic display that each attribute controls.



GDD Radial Object

Radial class information objects have additional attributes that are not shown in the figure.

For a detailed list of the attributes of radial class information objects, see Common Attributes of GDD Class Information Objects and Attributes of Radial Class Information Objects.

Running the Dynamic Display

After you customize the display, you must follow several steps before it can operate.

Setting up the Display

Before it can begin to display information, you must set up the dynamic display. GDD uses the class information object to provide the data needed to create the display.

To set up the dynamic display:

➔ Display the menu for the information object for your dynamic display and choose gdd setup.

GDD calls the procedure gdd-setup-class-definition to install the data from the class information object.

or

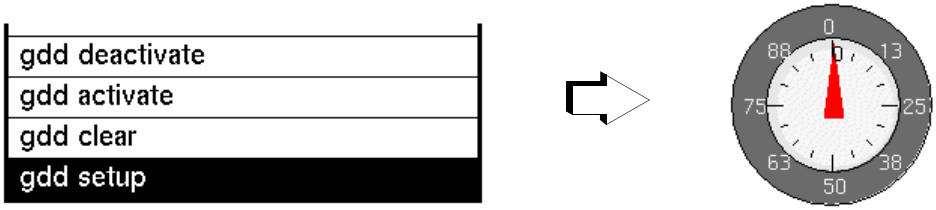
➔ gdd-setup-class-definition
(*info*: class gdd-class-information-object)

This procedure uses the gdd class information object provided to:

- Create a relation between the class definition and the class information object.
- Generate tick marks, labels, shadows, and tracking labels as desired.
- Activate the icon regions of the class definition specified in the information object.

Note If your display has more than one active region, you must set up each region separately, using the information object for each region.

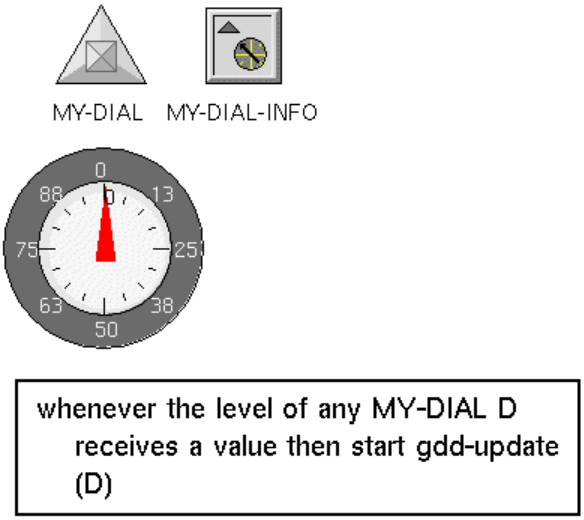
Your display should look like the one in the following figure:



Providing a Driver for the Display

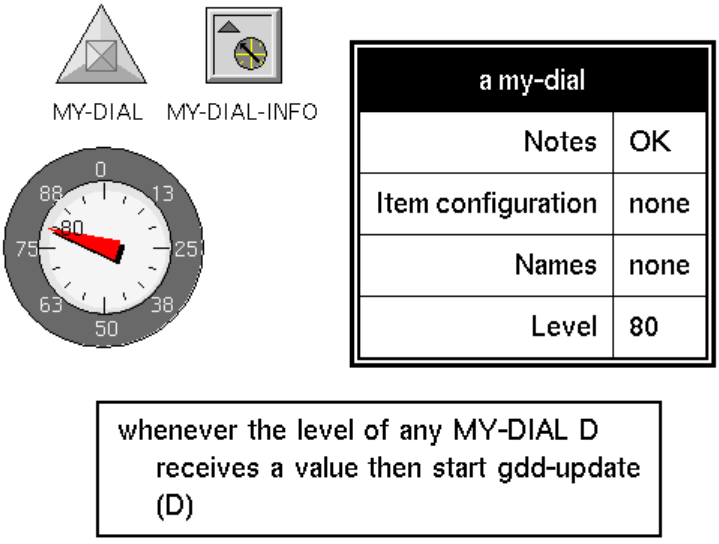
The display needs a driver to move the pointer. Your driver, either a procedure or a rule, must start the API procedure gdd-update. For more information on gdd-update see Updating Active Regions in the Icon.

The following figure shows the completed development workspace for the display and includes a rule to drive the display:



To test the dynamic display:

➔ Display the table of the dynamic display and modify the target attribute.



The pointer moves to the indicated value.

Clearing the Display

To remove the information associated with a dynamic display during set up:

➔ Display the menu for the information object for your dynamic display and choose gdd clear.

GDD calls the procedure gdd-clear-class-definition to remove the data from icon description.

or

➔ gdd-clear-class-definition
(*info*: class gdd-class-information-object)

This procedure clears the class definition by performing the following tasks:

- Deletes the relation between the class definition and the class information object.
- Removes the tick marks, labels, shadows, and tracking labels.
- Deactivates the icon regions of the class definition specified in the class information object.

Pausing and Restarting the Display

It is possible to pause a dynamic display while your KB continues to run, and to restart the display at a later time.

To pause a dynamic display:

➔ Display the menu for the information object for your dynamic display and choose gdd deactivate.

GDD calls the method gdd-deactivate to pause the display

or

➔ gdd-deactivate
(*info*: class gdd-class-information-object)

This method pauses the display the next time gdd-update is called.

Note Deactivating a display sets the pointer to the minimum-value position in the display.

To restart a dynamic display:

➔ Display the menu for the information object for your dynamic display and choose gdd activate.

GDD calls the method gdd-activate to restart the display.

or

➔ gdd-activate
(*info*: class gdd-class-information-object)

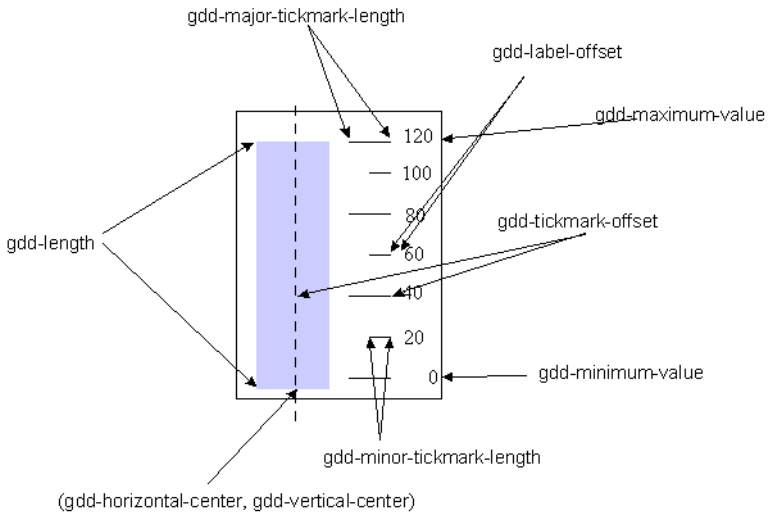
This method restarts a display that has been paused by gdd-deactivate.

Creating Linear and Floating Displays

You create Linear and floating displays in the same way as radial displays except, that they use **linear class information objects** and **floating class information objects**. Instead of specifying information specific to a radial display, you specify the following attributes:

- gdd-length: Specifies the length of the display in workspace units.
- gdd-alignment: Specifies whether the display is aligned horizontally or vertically.
- gdd-tickmark-offset: Specifies the distance between the center and the base of the tick marks.

The following figure summarizes the attributes of a linear class information object and the features of a linear dynamic display that each attribute controls. Floating class information objects have the same attributes. Both linear and floating class information objects have additional attributes that are not shown in the figure.



GDD Linear Object

For a detailed list of the attributes of linear class information objects, see Common Attributes of GDD Class Information Objects and Attributes of Linear and Floating Class Information Objects.

Suggestions to Minimize Flashing in a Display

If you design your own icon, you might observe some flashing in the background as the display changes. It is not unusual for power icons to flash as their active regions change. A technique exists to minimize and possibly eliminate this behavior. When drawing the pointer region, add an icon-background-layer to the icon. This is a layer that appears behind all other components of the icon and completely fills its background, regardless of the icon's size. If the color of this layer is set to the color of the icon layer that is directly beneath the pointer region, no flashing will occur when the icon is animated.

To specify an icon background color for a class:

1 Use the text editor to give the class definition's icon description an icon-background-layer section.

2 Specify the value of the layer to be either:

- Any G2 color
- A *region-name*

For example:

```
icon-background-layer: red
```

or:

```
icon-background-layer: lower-region
```

where lower-region is the name of the icon layer directly beneath the pointer region.

If this technique is not desirable, keep the pointer region reasonably small, with as few layers as possible. This helps to minimize flashing.

If you notice flashing when the dial is first animated, wait until the dial has changed values several times. Because of the way G2 icons are designed, flashing should be reduced each time the dial moves to a given position for the second time.

Cloning a Display From the GDD Library

The GDD Library contains a collection of dynamic displays representing typical displays requested by users. You can use any of these displays instead of designing your own.You can use a display exactly as it is displayed in the GDD library, or you can modify it by editing the attributes its class information object. If you clone a display directly from the GDD library and do not modify it, you do not need to have the development module, gdd-dev loaded.

To use a display as it is shown in the GDD library:

1 Clone the display from the library and place it on your workspace.

2 On the table for the display, note the name of the attribute being displayed.

3 Conclude a value into the attribute being displayed.

The display is updated.

If you wish to modify the display, you must have the module gdd-dev loaded.

To modify a display from the GDD Library:

1. Clone the class definition of display you wish to use.
2. Give the class definition a name.
3. Display the subworkspace of the class definition.

   The subworkspace of the class definition contains the information object for the class.
4. On the table for the class information object, change the gdd-target-class attribute to specify the name of the cloned class definition.
5. Change the values of any attributes you wish to modify.
6. Choose gdd setup from the menu of the class information object.

   This will make the association between the newly cloned class and the display.
7. Add a driver for the display.

   The easiest way to do this is to modify the rule which is in the subworkspace with the class information object.

Removing the Development KBs

Once you have completed all of your dynamic displays, you can remove the gdddev and gddlib modules from your application. The only module needed to run the dynamic displays is gddroot. Use the API procedures provided by gddroot to manage your dynamic displays.

To delete the gdddev or gddlib module:

1. Choose Main Menu > Miscellany > Delete Module to display a menu of the modules currently loaded in G2.

   For example:
2. Choose gdddemo.

   The following dialog appears:
3. Click All on the dialog that appears to delete the module and all its associated workspaces.

4

GDD Class Information Objects

Describes each attribute and its value for the three kinds of class information objects

Introduction

The GDD class information object is maintained on the subworkspace of the class definition. The attributes of a class information object specify the runtime behavior of the instances of the associated class. Most of the attributes apply to all classes of information objects. In addition, there are specific attributes for the radial class and shared attributes for the floating and linear classes.

This chapter lists each attribute and describes the properties of the dynamic display each specifies.

Common Attributes of GDD Class Information Objects

Attribute	Description
gdd-status	(Read Only) Indicates if the display is active or deactivated.
Allowable values:	
Default value:	none
Notes:	This is not a user settable attribute. It is designed to be set by GDD. Possible values are none, active, deactivated.
gdd-target-class	Specifies the class with which this display is associated.
Allowable values:	The name of any valid GDD class
Default value:	none
gdd-target-attribute	Specifies the attribute that is displayed.
Allowable values:	Any valid attribute name
Default value:	none
gdd-minimum-value	Specifies the minimum value on the display.
Allowable values:	Any quantity
Default value:	0
gdd-maximum-value	Specifies the maximum value on the display.
Allowable values:	Any quantity
Default value:	100

Attribute	Description
gdd-horizontal-center	Specifies the x position in workspace units of the center of the display.
Allowable values:	Any positive integer
Default values:	100 for radial display 50 for linear and floating display
gdd-vertical-center	Specifies the y position in workspace units of the center of the display.
Allowable values:	Any positive integer
Default values:	100 for radial display 50 for linear and floating display
gdd-tickmark-count	Specifies the number of tick marks in the display.
Allowable values:	Any value between the gdd-minimum-value and the gdd-maximum-value of the display
Default value:	16 for radial display 17 for linear and floating display
Notes:	tick marks specify intermediate values between the gdd-minimum-value and the gdd-maximum-value or between major tick marks.
gdd-tickmark-color	Specifies the color of the tick marks.
Allowable values:	Any G2 color
Default value:	black
gdd-tickmark-length	Specifies the length of the tick marks in workspace units.
Allowable values:	Any positive integer

Attribute	Description
Default value:	3
gdd-major-tickmark-count	Specifies the number of major tick marks in the display.
Allowable values:	The integer 0 or any integer between 2 and 100
Default values:	8 for radial display 9 for linear and floating display
Notes:	Major tick marks specify intermediate values between the gdd-minimum-value and the gdd-maximum-value of the display. A value of 0 indicates there are no major tick marks.
gdd-major-tickmark-length	Specifies the length of the major tick marks in workspace units.
Allowable values:	Any positive integer
Default value:	6
gdd-label-count	Specifies the number of labels that appear on the display.
Allowable values:	The integer 0 or any integer between 2 and 100
Default value:	9
Notes:	A label appears next to a tick mark and indicates the value associated with the tick mark. A value of 0 indicates that there are no labels.
gdd-label-color	Specifies the color of the labels.
Allowable values:	Any G2 color
Default value:	black

Attribute	Description
gdd-label-size	Specifies the size of the font used in the labels.
Allowable values:	small, large, extra-large
Default value:	small
gdd-label-offset	Specifies the distance in workspace units from the base of a tick mark and the first character in its label.
Allowable values:	Any integer between 0 and 500
Default value:	25
gdd-shadow	If true, specifies that the pointer in a radial or floating display or the active region in a linear display has a shadow.
Allowable values:	true, false
Default value:	true
gdd-shadow-offset	When gdd-shadow is true, specifies that the pointer region is shadowed on the display.
Allowable values:	Any integer from 0 to 10
Default value:	3
Notes:	Specifies the thickness of the shadow on the pointer.
gdd-shadow-color	When gdd-shadow is true, specifies the color of the shadow.
Allowable values:	Any G2 color
Default value:	dim-gray

Attribute	Description
gdd-tracking-label	Specifies that the display has a moving label that tracks the active region and gives the current value of the target attribute.
Allowable values:	true, false
Default value:	false
gdd-tracking-label-color	When gdd-tracking-label is true, specifies the color of the tracking label.
Allowable values:	Any G2 color
Default value:	black
gdd-tracking-label-size	When gdd-tracking-label is true, specifies the font size of the tracking label.
Allowable values:	small, large, extra-large
Default value:	small
gdd-tracking-label-offset	When gdd-tracking-label is true, specifies the distance in workspace units from the first character of the tracking label to the center of the display.
Allowable values:	Any integer between 0 and 500
Default value:	30
gdd-pointer-region	Specifies in a sequence the active regions of the display.
Allowable values:	

Attribute	Description
Default value:	sequence()
Notes:	This sequence defines the pointer or active region in a GDD display. For example: (the symbol pointer-1)

Attributes of Radial Class Information Objects

Attribute	Description
gdd-radius	Specifies the radius in workspace units from the center of the display to the base of the tick marks.
Allowable values:	An integer between 0 and 500
Default value:	54
Notes:	An icon can be no larger than 500 X 500 workspace units. A semi-circular display whose radius is 500 workspace units is allowable, but a circular display can have a radius of no more than 250 units so that its diameter remains under the allowable limits for icons.
gdd-range	Specifies the size of the circular arc, in degrees, making up the radial display.
Allowable values:	Any quantity from 0.0 to 360.0
Default value:	360.0
Note:	The default display is a full circle.
gdd-start-point	Specifies the initial position of the pointer, measured in degrees, clockwise, from the vertical.
Allowable values:	Any quantity from 0.0 to 360.0
Default value:	0.0
Notes:	Both 0.0 and 360.0 define a upward pointing vertical pointer. With a value of 90.0 degrees, the pointer points toward the right; with 270.0 degrees, the pointer points leftward.

Attribute	Description
gdd-clockwise	Specifies that the pointer travels in a clockwise direction.
Allowable values:	true, false
Default value:	true

Attributes of Linear and Floating Class Information Objects

Attribute	Description
gdd-length	Specifies the length of the display in workspace units.
Allowable values:	Any positive integer from 0 to 500.
Default value:	160
Notes:	The maximum length of the display is determined by the maximum allowable size for an icon in G2, which is 500 X 500 workspace units
gdd-alignment	Specifies the orientation of the display.
Allowable values:	vertical, horizontal
Default value:	vertical
gdd-tickmark-offset	Specifies the distance in workspace units between the horizontal center (for vertical displays) or the vertical center (for horizontal displays) and the base of the tick marks.
Allowable values:	An integer between 0 and 500
Default value:	40

5

The GDD API Reference

Describes all supported GDD API calls.

Introduction

The GDD API is designed to allow developers of G2 applications to take full advantage of power icons. The interface is usable at many levels. Users can use the high level routines to create custom gauges and dials, or use lower level routines to create and activate other kinds of displays. The procedures are listed in functional groups.

Manipulating Class Definitions

gdd-setup-class-definition

(*info*: class gdd-class-information-object)

Uses the GDD class info object provided to:

- Create a relation between the class definition and the gdd-class-information-object.
- Generate tick marks, labels, shadows, and tracking labels, as desired.
- Activate the icon regions of the class definition specified in the info object.

gdd-clear-class-definition

(*info*: class gdd-class-information-object)

Clears the class definition by performing the following tasks:

- Deletes the relation between the class definition and the gdd-class-information-object.
- Removes the tick marks, labels, shadows, and tracking labels.
- Deactivates the icon regions of the class definition specified in the info object.

gdd-clear-all-gdd-info

(*def*: class class-definition)

Clears all GDD created layers and variables from a class definition. This procedure can be used to clear a class definition if one or more of the gdd-class-information-objects has been deleted.

gdd-get-class-variables

(*obj*: class item)

-> *variables*: structure

Returns a structure containing the variables of a class along with their initial values.

gdd-activate

(*def*: class class-definition)

A method that re-activates a class definition that has been deactivated using gdd-deactivate.

gdd-deactivate

(*def*: class class-definition)

A method that deactivates a class definition.

Activating and Deactivating Icon Regions

gdd-make-icon-active

(*def*: class class-definition)

Makes all named regions in the icon active.

gdd-make-icon-layer-active

(*icon-desc*: structure, *layer-name*: symbol)
-> *new-desc:* structure

Accepts the icon description expressed as a structure and makes the named layer in the icon active. Returns the icon description.

gdd-make-icon-inactive

(*def*: class class-definition)

Makes all named regions in the icon inactive.

gdd-make-icon-layer-inactive

(*icon-desc*: structure, *layer-name*: symbol)
-> *new-desc:* structure

Accepts the icon description expressed as a structure and makes the named layer in the icon inactive. Returns the icon description.

Updating Active Regions in the Icon

gdd-update

(*def*: class class-definition)

Updates the positions of the active regions in the icon, based on the named attribute.

Moving Regions in the Icon

gdd-rotate-icon

(*def*: class class-definition, *degrees*: float, *x-origin*: integer, *y-origin*: integer)

Rotates all of the layers in the icon by the number of degrees indicated.

gdd-rotate-icon-layer

(*layer-name*: symbol, *variables:* structure*, degrees*: float, *x-origin*: integer, *y-origin*: integer)
-> *new-vars:* structure

Rotates the named icon layer by the number of degrees indicated and returns the updated set of icon variables.

gdd-shift-icon

(*def*: class class-definition, *x-shift*: integer, *y-shift*: integer))

Shifts all of the icon layers by the number of workspace units indicated.

gdd-shift-icon-layer

(*layer-name*: symbol, *variables:* structure*, x-shift*: integer, *y-shift*: integer)
-> *new-vars*: structure

Shifts the named icon layer by the number of workspace units indicated and returns the updated set of icon variables.

Activating and Deactivating Class Information Objects

gdd-activate

(*info*: class gdd-class-information-object)

A method that reactivates a class information object that has been deactivated using gdd-deactivate.

gdd-deactivate

(*def*: class gdd-class-information-object)

A method that deactivates a class information object.

Glossary

A

active region: A region of an icon whose coordinates are defined by icon variables.

C

Class Information Objects: GDD objects that contain in their tables information about the properties of a dynamic display. These properties include: the class of object whose attributes are displayed, the attribute of that object to display, the range of values on the display, the number of and length of the tick marks on the display, the labels on the display, the appearance and movement direction of the animated part of the display, and others.

D

dynamic displays: Animated graphical attribute readouts, dynamically generated, that can be added to icons in your application.

F

floating class information objects: A class information object whose attributes contain specific information about floating displays.

floating display: A type of linear display that has an active region that shifts position.

G

G2 Dynamic Displays Utility: A G2 Utility that gives you the ability to add dynamic displays to icons in your applications.

GDD: The acronym for G2 Dynamic Displays.

GDD Demo: A KB of examples that cover the main features of GDD.

GDD Development: The Development component of GDD. Contains the tools for developing new GDD displays.

GDD Library: The library component of GDD. Contains samples of the three basic kinds of displays. Each can be easily customized for a particular application by using the icon editor and by editing the attributes.

GDD Root: The runtime component of GDD. Used when you are running applications that include dynamic displays. Provides access to a list of the signatures of the API procedures necessary to manage dynamic displays after they are developed.

I

icon variables: User-defined variables that specify almost any element of a class's icon description. Using the conclude action, you can change the value of an icon variable in an instance of the class. The instance's icon immediately changes to reflect the new value.

L

linear class information objects: A class information object whose attributes contain specific information about linear displays.

linear display: A rectangular shaped display that has an active region that changes length to indicates a value.

P

palette: A collection of objects that can be cloned for use in application development. The GDD palette contains the three kinds of class information objects.

R

radial class information object: A class information object whose attributes contain specific information about linear displays.

radial display: A display that has in its active region, a pointer that rotates to indicate a value.

T

tick marks: Small lines on a display representing intermediate positions between the maximum value and the minimum value on the display.

tracking label: A label near the animated part of the display that displays the current value of the target attribute.

Index

I

L

M

O

P

R

S

W