G2 Relation Browser

User's Guide

Version 2.3 Rev. 0

G2 Relation Browser User's Guide, Version 2.3 Rev. 0

May 2007

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

Part Number: DOC007-230

Contents

Preface

Describes this guide and the conventions that it uses.

About this Guide

This guide describes the G2 Relation Browser (GRLB) module. This module provides the ability to display G2 relations and user-defined relations in a graphical layout. SymCure, which is available as part of the Optegrity bundle, makes use of GRLB to display fault models graphically.

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
<u>*text-string*</u>	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 <u>underlined</u>. 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)
–> <u>*transferred-items*</u>: 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`

Introduction to the G2 Relation Browser

Describes the G2 Relation Browser (GRLB) module.

Introduction

The G2 Relation Browser (GRLB) provides the ability to display G2 relations and user-defined relations in a graphical layout. For example, SymCure, a component of Optegrity and Integrity, is a graphical modeling language used to build cause-and-effects models and makes use of GRLB for displaying relations in generic and specific event models. For more information, see the *SymCure User's Guide*.

The G2 Relation Browser displays all relations or a selected relation of an object. The browser provides three different layouts: default (columnar), circular, or breadth-first. The default layout is arranged so that the relations are listed in a column and graphically connected to the target object. The circular layout places the relations in a circle around the target object and graphically connected to the target object. The breadth-first layout can display related objects, based on a hierarchy, for example, a workspace hierarchy or domain map.

The connections are color-coded to group related relations visually. In the default and circular views, the relation name appears on the connection half-way between the target object and the related object.

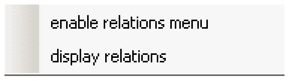
You can also create your own layouts and relation-collection methods. This feature allows you to customize the browser to display any type of relation with any type of layout.

Loading GRLB

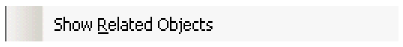
To use the GRLB module, you must load or merge in `grlb.kb`, which is located in the `g2i\kbs` directory.

Menu Choices

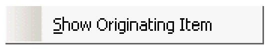
The G2 Relation Browser adds the following menu choices to objects:



GRTL items provide the following popup menu choice on items when the G2 Relation Browser is loaded, which is the equivalent of the display relations menu choice:



GRTL items in a Relation Browser also provide the following popup menu choice:

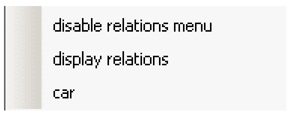


Displaying All Relations

To display all relations for an object, select the display relations menu choice. Selecting this menu choice creates a workspace with a header labeling the selected object. The object for which the relations are shown is displayed with connections coming from it and going to the item to which it is related. The connections are color-coded to group similar relations. The relation name also appears on the connection and between the two related items.

Displaying a Selected Relation

To display a single relation, select the enable relations menu choice to activate additional user menu choices for the object, which are the relations of the selected object. Below is an example of the popup menu for an object after selecting the enable relations menu choice, which has one relation named car:

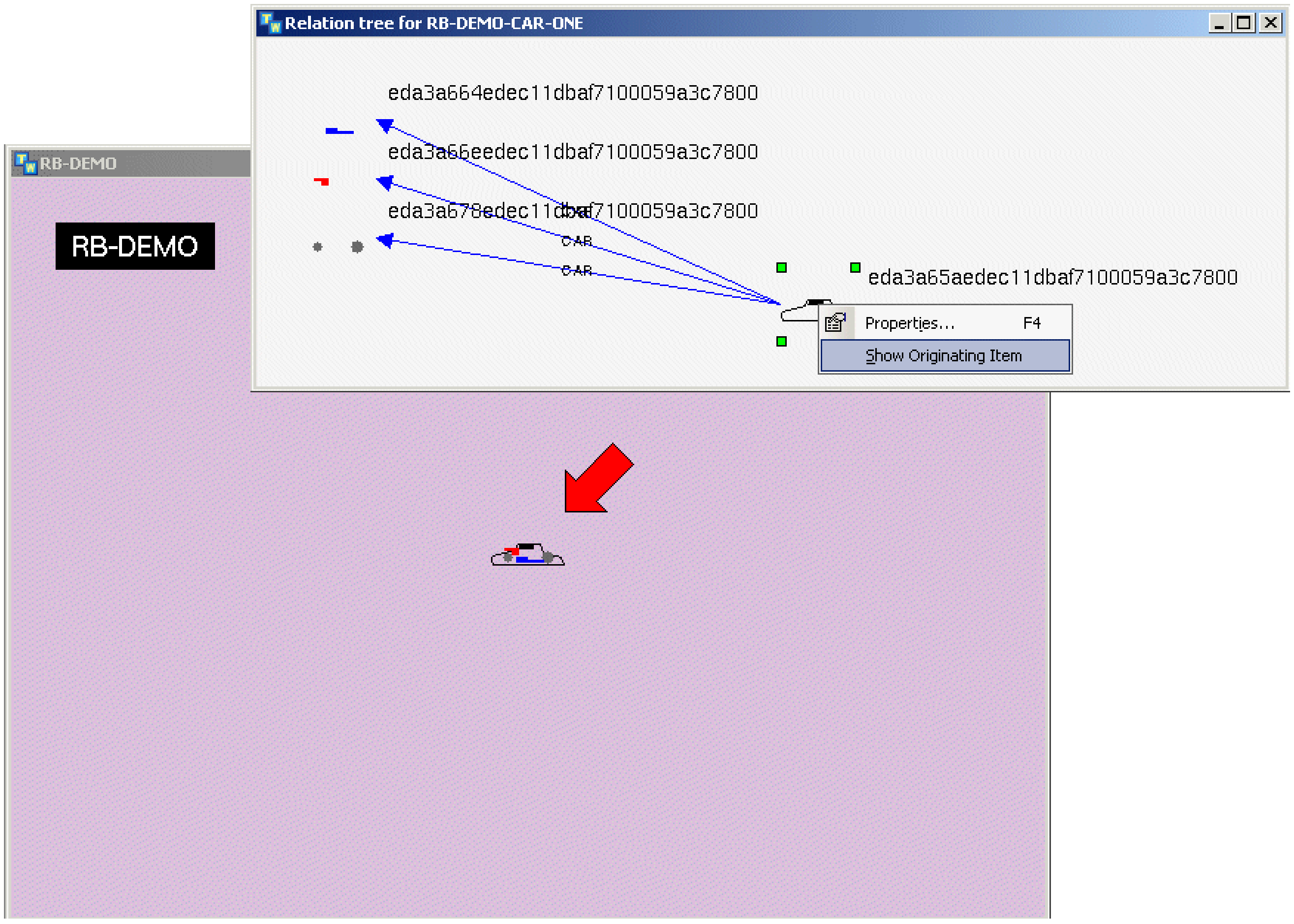


Selecting the additional menu choice displays the selected object and its relations to other objects.

The disable relations menu menu choice removes the user menu choices that the enable relations menu choice adds.

Showing the Originating Item

When you display the Relation Browser, you can choose Show Originating Item on any proxy item in the browser to go to the original item. For example:


Relation tree for RB-DEMO-CAR-ONE
eda3a664edec11dbaf7100059a3c7800
eda3a66eedec11dbaf7100059a3c7800
eda3a678edec11dbaf7100059a3c7800
CAR
CAR
eda3a65aedec11dbaf7100059a3c7800
Properties... F4
Show Originating Item
RB-DEMO
RB-DEMO

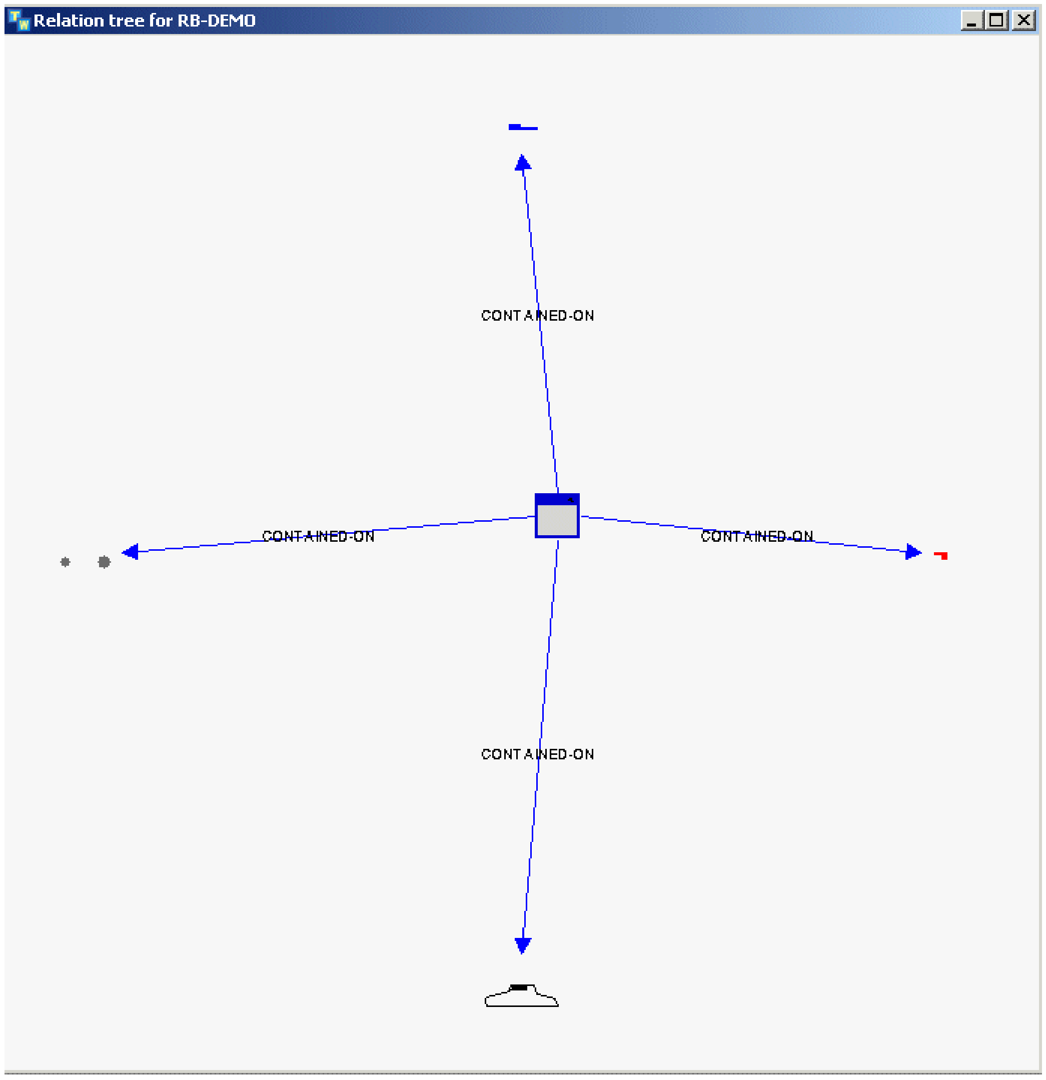
G2 Relation Browser

The G2 Relation Browser has two layouts of the Relation Browser: a circular view and a vertical view.

To configure which layout GRLB uses, you configure a parameter in the configuration file. For details, see Chapter 2, “Configuration File” on page 7.

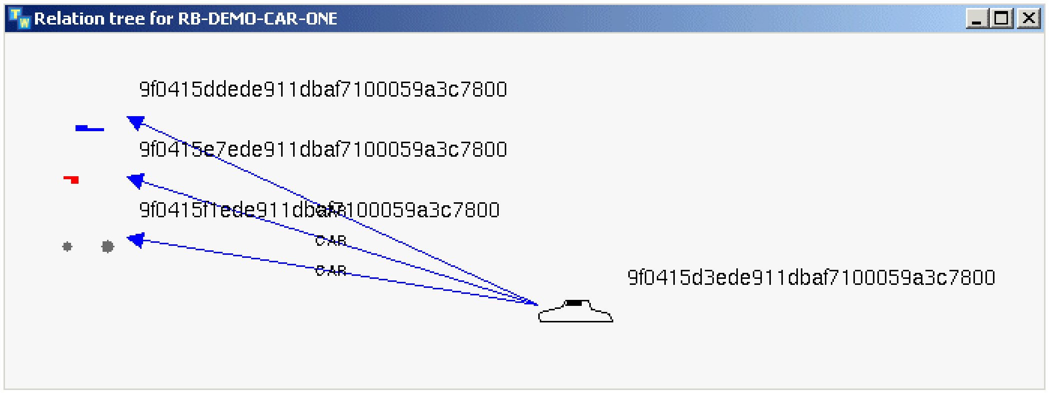
Circular View

The circular view displays the related objects around the selected object. The related objects are connected with color coding to distinguish the different types of relations. Below is an example of a circular view:



Vertical View

The vertical view displays relations in a vertical column to the left of the selected object. The connections are color-coded and include the relation name. Here is an example that uses the vertical layout:


Relation tree for RB-DEMO-CAR-ONE
9f0415ddede911dbaf7100059a3c7800
9f0415e7ede911dbaf7100059a3c7800
9f0415f1ede911dbaf7100059a3c7800
CAR
CAR
9f0415d3ede911dbaf7100059a3c7800

Configuration File

Describes the G2 Relation Browser (GRLB) configuration file settings.

Introduction

This table describes the settings in the configuration file (`config.txt`, by default) for the GRLB module:

Group	Configuration File Settings	Description
GRLB	`GRLB-DEFAULT-LAYOUT-ALGORITHM=grlb-circular-layout`	Controls the layout for the display of relations. By default, the layout is circular, that is, the selected item is shown at the center of a circle and its related items are shown along the circumferance of the circle. The radius of the circle is determined by `_GRLB-RADIUS`. The other option is `grlb-default-layout`, which shows the related objects above the selected item.
GRLB	`_GRLB-RADIUS=400`	When using a circular layout, the radius of the circle.

Group	Configuration File Settings	Description
GRLB	`GRLB-DELETE-PROXY-CLASS-DEFINITIONS-ON-STARTUP=true`	By default, proxy class definitions are deleted whenever there is a restart. Set this parameter to `false` to cache proxy class definitions within repositories in the application KB, in which case they are never deleted.

G2 Relation Browser API

Describes the G2 Relation Browser (GRLB) API procedures.

Introduction

Included in the G2 Relation Browser is an application programmers' interface (APIs). These APIs allow you to create customized displays and customized relation collection methods. This section first describes the customization of the built-in layouts, then it describes the customization for the relation collector, and finally it describes how to create a new layout.

Layout Objects

The G2 Relation Browser includes two different layouts, circular and vertical. These are defined in the form of an object. These objects contain attributes that are used specifically for their respective layout algorithms.

The vertical layout object, grlb-layout-default, contains the following attribute:

Attribute Name	Description
connection-length	The distance in workspace units between the originating object and its related objects.

The circular layout object, grlb-layout-circular, contains the following attribute:

Attribute Name	Description
radius	This attribute is used as the radius value for drawing the originating object and its related objects.

You can subclass these objects for customization. If you subclass these objects, you must provide the drawing methods for the new layout. See “Relations Collection” on page 10 and “Creating New Layouts” on page 11.

Relations Collection

The default behavior of the Relation Browser is to obtain and display all G2 relations for a given object instance. The method item::grlb-get-relations collects the G2 relations. You can overload this method to customize it.

Here is the API for the grlb-get-relations method:

grlb-get-relations
(*Item*: class item *SearchRelName*: symbol, *Client*: class object)
–> *relations*: sequence

Attribute Name	Description
Item	The object instance for which relations should be collected.
SearchRelName	Used to return only relations defined by the value of this parameter. For example, if SearchRelName is set to the symbol connected-to, this method would only return relation information pertaining to the connected-to relation.
Client	Reserved for future use.

This method returns a sequence of structures. The structure represents the collected information on the relation and has this syntax:

structure
(rel-name: *symbol*,
related-items: *sequence*)

Structure Attribute	Description
rel-name	The relation name.
related-items	A sequence containing the related items.

You can override the grlb-get-relations method; however, you cannot change the number of arguments. This method is also expected to return a sequence of structures as outlined above.

Creating New Layouts

To create a new layout, you must define a new layout object and create a grlb-draw-relations method.

Existing Layouts

There are two built-in layouts, grlb-layout-default and grlb-layout-circular, which are described above.

Creating a Layout Object

You can subclass from either of the two existing layout objects or subclass from their parent class _grlb-layout-object. You can add attributes to the subclass, as needed, for the new layout. For example, the grlb-layout-circular class contains the radius attribute. You can also reference these attributes in your custom grlb-draw-relations method.

Creating the grlb-draw-relations Method

To activate your new drawing layout, you must implement the grlb-draw-relations, whose signature is:

grlb-draw-relations
    (*Layout*: class grlb-layout-XXXX, *Item*: class item,
    WS: class kb-workspace, *Relations*: sequence, *Client*: class ui-client-item)

Attribute Name	Description
Layout	The newly created layout object.
Item	The item to which the relations belong.
WS	The workspace that will contain the drawn items.
Relations	The sequence of structures created by your grlb-get-relations method.
Client	The window in which the workspace is displayed.

You can view the grlb-layout-circular::grlb-draw-relations or the grlb-layout-default::grlb-draw-relations methods as an example and starting point.

Displaying Relations Programmatically

Once you have created a new layout object definition and have implemented the grlb-get-relations method and the grlb-draw-relations method, you can use the grlb-show-relations procedure to draw the relations gathered by the grlb-get-relations method. They are displayed according to your new layout and the grlb-draw-relations method.

The API for the grlb-show-relations is:

grlb-show-relations
 (*Layout*: class _grlb-layout-object, *Item*: class item, *RelName*: symbol, *Client:* class ui-client-item)

Attribute Name	Description
Layout	The layout object that is responsible for drawing the relations.
Item	The item to which the relations belong.
RelName	A symbol that is passed to the grlb-get-relations method for collection.
Client	The window in which the workspace is displayed.

The grlb-show-relations procedure calls grlb-create-workspace, described below. If you choose to develop your own grlb-show-relations procedure, remember to call this procedure to create the workspace. You must also remember to show the workspace on the client item that is passed to the grlb-show-relations procedure.

The display relations menu choice on an object activates the grlb-show-relations to collect and draw the relations for the select object.

APIs

grlb-show-relations
(*Layout*: class _grlb-layout-object, *Item*: class item, *RelName*: symbol, *Client*: class ui-client-item)

grlb-create-workspace
(*Item*: class item, *Client*: class ui-client-item)
–> *WS*: class kb-workspace

grlb-get-relations
(*Item*: class item, *SearchRelName*: symbol, *Client*: class object)
–> *Relations*: sequence

See "Creating New Layouts" on page 11 for information on overloading.

grlb-draw-relations
(*Layout*: class grlb-layout-default, *Item*: class item,
WS: class kb-workspace, *Relations*: sequence, *Client*: class ui-client-item)

See "Creating New Layouts" on page 11 for information on overloading.

grlb-is-item-proxy-object
(*Item:* class item)
–> *result:* truth-value

Returns true if *Item* is a grlb-proxy-item; otherwise, returns false.

grlb-delete-proxy-class-definitions
( )

Deletes every transient grlb-tree-workspace and every proxy class definition contained on the subworkspace of any grtl-class-instance-holder that is contained in the top level module of the application. This API is called in GRLB's startup procedure to automatically delete proxy class definitions from the top-level application module at application startup. Note that this procedure deletes most grlb-workspaces, including SymCure specific fault model displays before deleting proxy class definitions.

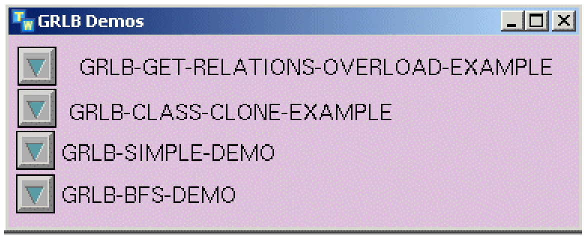
G2 Relation Browser Demo

Describes the G2 Relation Browser (GRLB) demo.

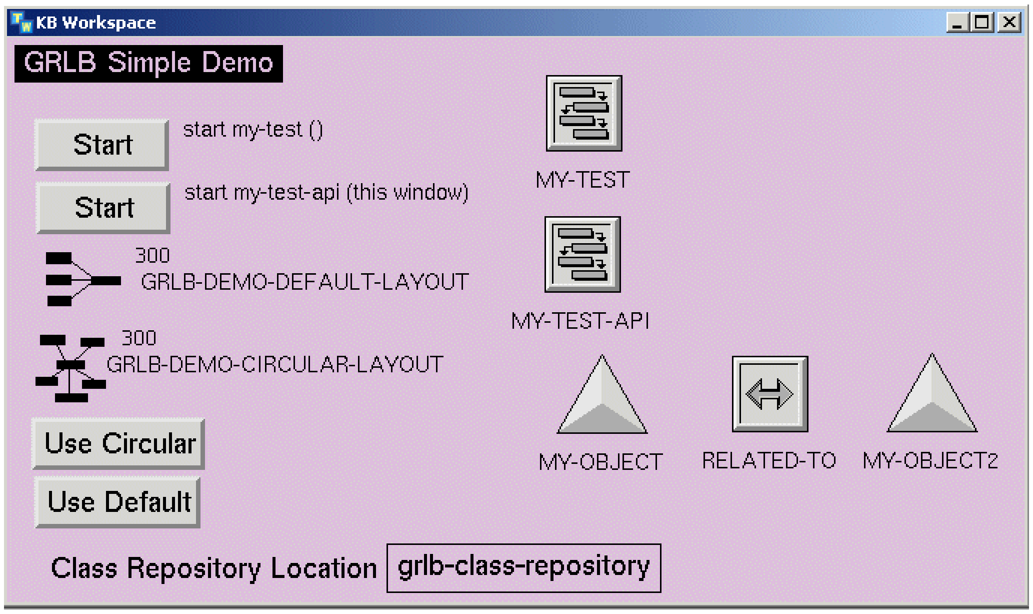
Introduction

The G2 Relation Browser Demo demonstrates the capabilities of the G2 Relation Browser.

To access the demos, click the Demos button on the top-level workspace to display this workspace:



Go to the grlb-simple-demo workspace:

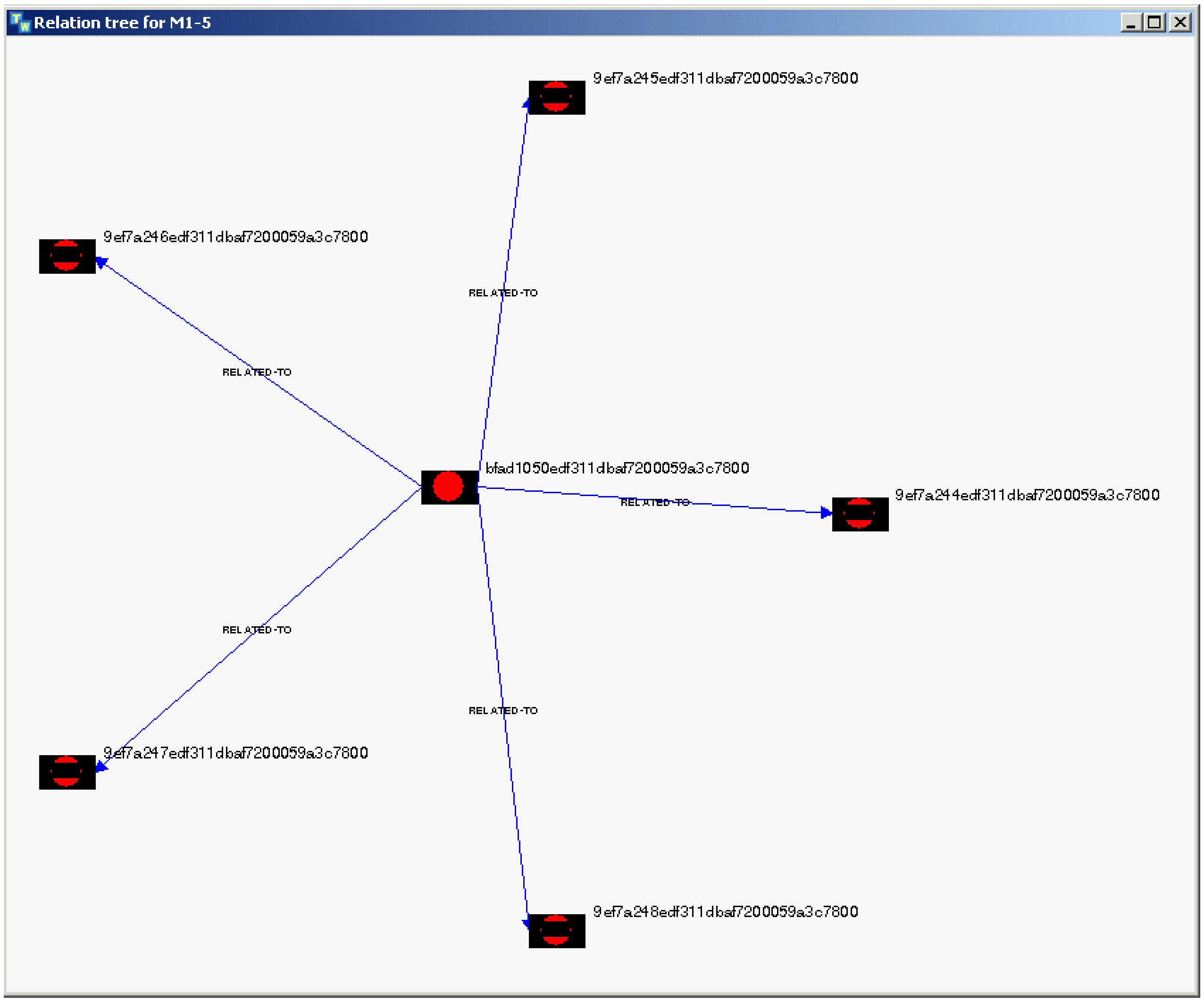


Starting the Demo

The demo workspace has two Start buttons. The button labeled start my-test creates five instances of my-object and five instances of my-object2. These instances are displayed in the middle of the demo workspace. After these objects are created, the related-to relation is created between the my-object and my-object2 instances.

Displaying the Relations

To display the relations created by the my-test procedure, select any of the instances created and choose display relations to display this workspace:

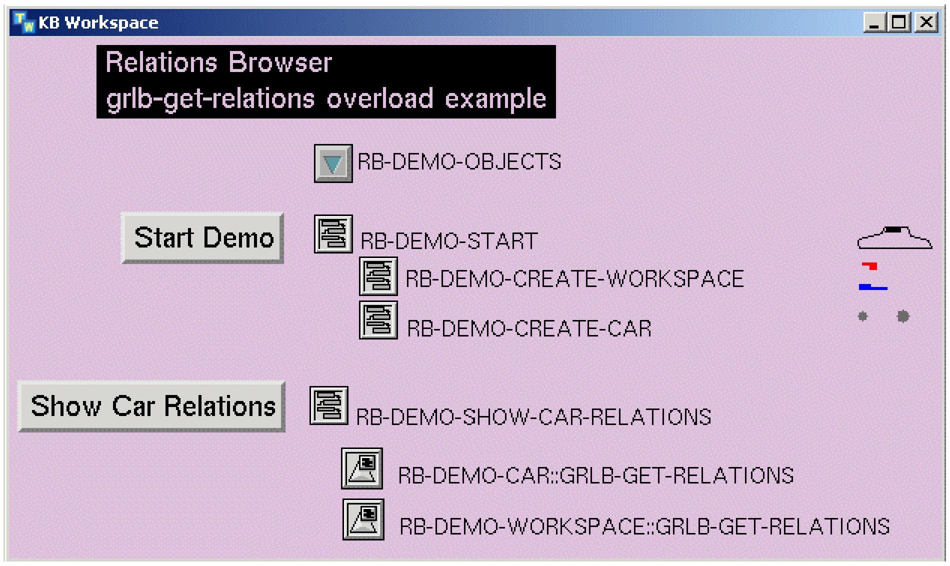

Relation tree for M1-5
9ef7a245edf311dbaf7200059a3c7800
9ef7a246edf311dbaf7200059a3c7800
RELATED-TO
RELATED-TO
bfad1050edf311dbaf7200059a3c7800
RELATED-TO
9ef7a244edf311dbaf7200059a3c7800
RELATED-TO
RELATED-TO
9ef7a247edf311dbaf7200059a3c7800
9ef7a248edf311dbaf7200059a3c7800

Demonstrating the API

The Start button labeled start my-test-api (this window) creates the same objects as before, creates the relations between the my-object and my-object2, but it also calls the grlb-show-relations API. The my-test-api procedure makes two calls to the grlb-show-relations API. The first call passes the grlb-demo-default-layout object and the second call passes the grlb-demo-circular-layout object; this is the only difference between these two API calls. When these calls are made, the workspaces displayed are the same as above.

API Overload Example

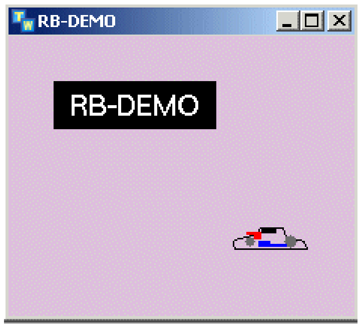
On the demos workspace, go to the grlb-get-relations-overload-example workspace:


KB Workspace
Relations Browser
grlb-get-relations overload example
RB-DEMO-OBJECTS
Start Demo
RB-DEMO-START
RB-DEMO-CREATE-WORKSPACE
RB-DEMO-CREATE-CAR
Show Car Relations
RB-DEMO-SHOW-CAR-RELATIONS
RB-DEMO-CAR::GRLB-GET-RELATIONS
RB-DEMO-WORKSPACE::GRLB-GET-RELATIONS

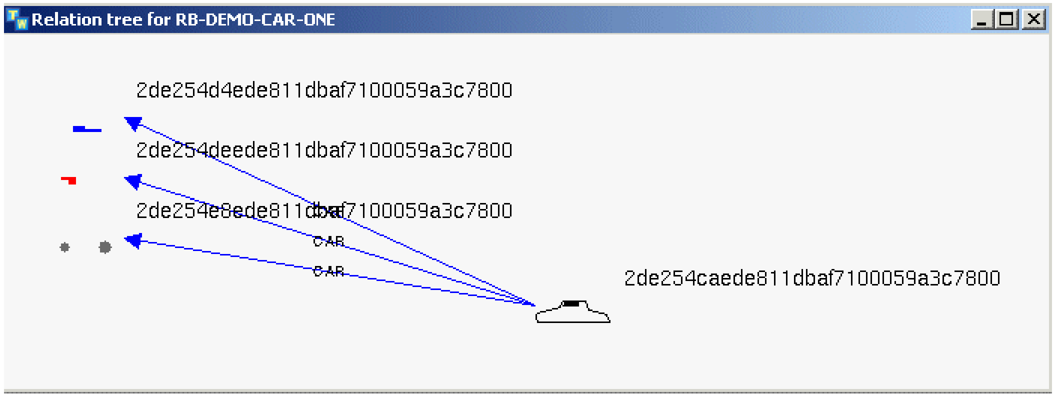
Starting the Overload Demo

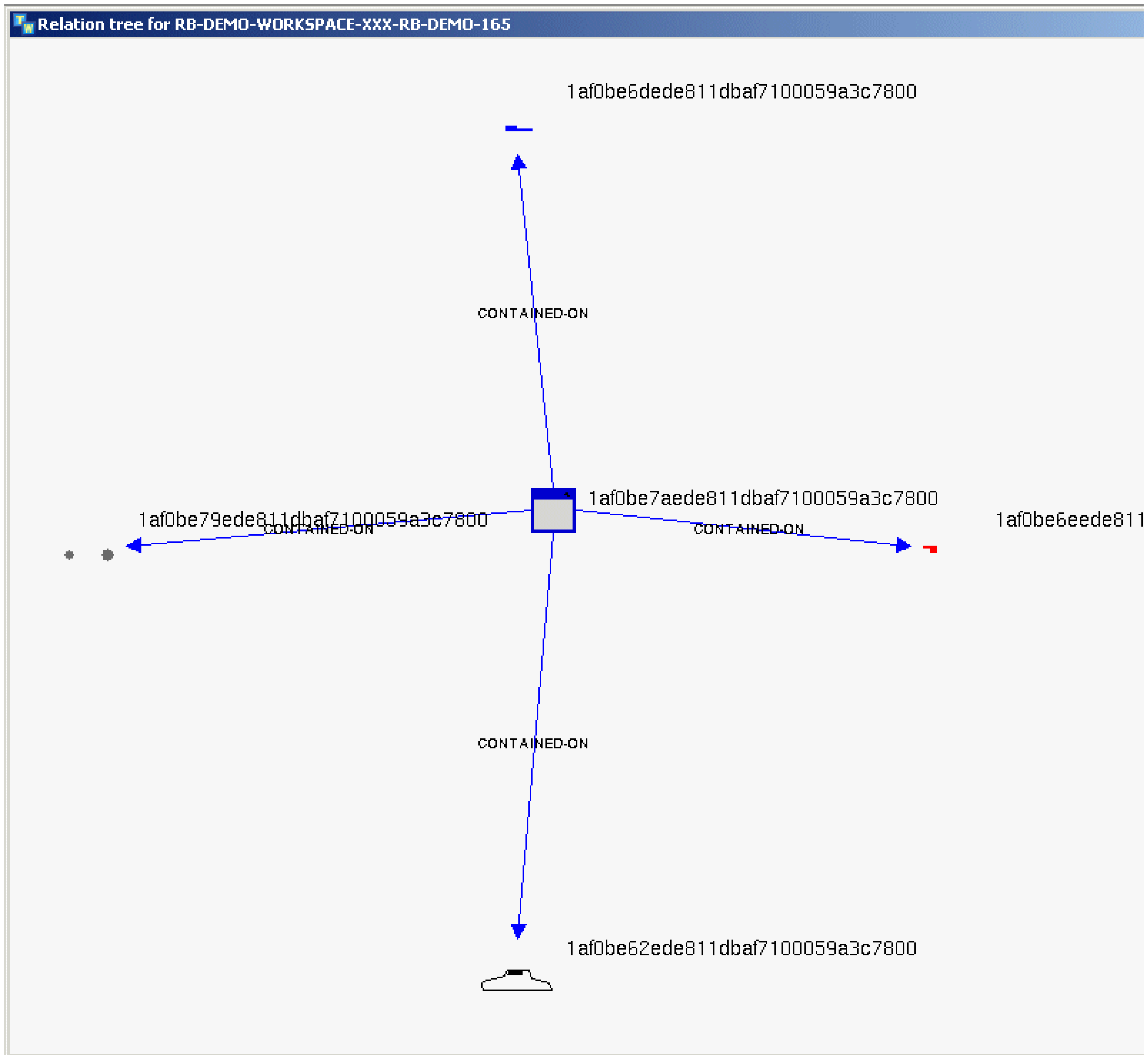
This workspace has two buttons labeled Start Demo and Show Car Relations. Selecting the Start Demo button creates a workspace, then creates components of a car. In this example, no G2 relations are created; however, during the creation of the car components, the symbol ONE is added to the end of every components name.

The following workspace appears after clicking the Start Demo button:



In this example, there are two sets of relations that can be derived, first is the relation of all the objects on the workspace; second, is the virtual relation of all the objects containing the symbol ONE at the end of their name. The Show Car Relations button displays two layouts. The default layout displays the objects related by the symbol ONE at the end of the objects name. The circular layout displays the objects contained on the rb-demo workspace. Below are the resulting workspaces.


Relation tree for RB-DEMO-CAR-ONE
2de254d4ede811dbaf7100059a3c7800
2de254deede811dbaf7100059a3c7800
2de254e8ede811dbaf7100059a3c7800
CAR
CAR
2de254caede811dbaf7100059a3c7800


Relation tree for RB-DEMO-WORKSPACE-XXX-RB-DEMO-165
1af0be6dede811dbaf7100059a3c7800
CONTAINED-ON
1af0be79ede811dbaf7100059a3c7800
CONTAINED-ON
1af0be7aede811dbaf7100059a3c7800
CONTAINED-ON
1af0be6eede811
CONTAINED-ON
1af0be62ede811dbaf7100059a3c7800

Overload Example Code

To see the code used to create this overloading example, go to the grlb-get-relations-overload-example workspace and view following procedures and methods:

- rb-demo-show-car-relations
- rb-demo-car::grlb-get-relations
- rb-demo-workspace::grlb-get-relations

Index