G2 Run-Time Library

User's Guide

Version 2.3 Rev. 0

G2 Run-Time Library User's Guide, Version 2.3 Rev. 0
September 2016

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: DOC132-230

Contents Summary

Contents

Preface

Describes this guide and the conventions that it uses.

About this Guide

This guide describes the G2 Run-Time Library (GRTL) module. This module provides a wide variety of development tools for the runtime environment. These include support for object models, which includes object keys, event notification, and support for localization, configuration files, command-line options, and a variety of general runtime utilities.

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 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*

Part I

Introduction

Chapter 1: Introduction to the G2 Run-Time Library

Describes the G2 Run-Time Library (GRTL) module.

Chapter 2: Module Settings

Describes module settings to configure operations of GRTL procedures and other modules relying on GRTL.

Chapter 3: KB Initialization

Describes KB initialization procedures.

1

Introduction to the G2 Run-Time Library

Describes the G2 Run-Time Library (GRTL) module.

Introduction

G2 Run-Time Library (GRTL) is a core module that defines a common set of utility functions, an object model, and user interface patterns to speed up the development process and provide a consistent object model. This framework is used by several Gensym application products such as ReThink, e-SCOR, and Optegrity, but you can also use it to quickly build custom applications. When used in combination with GEVM, it also provides message handling and message browsing capabilities, which are common in many applications.

GRTL provides an infrastructure for addressing key issues within a common framework, including:

- Utility functions and procedures

  GRTL provides numerous utility functions to improve productivity and build consistency in applications.

- Application error handling

  GRTL relies on the G2 Error Handling (GERR) functionality to provide extended error handling and logging.

- Application configuration

  GRTL includes APIs to extract application configuration from external text configuration files, in addition to providing KB-level configuration of modules.

- Application localization

  GRTL includes APIs to localization text into multiple languages with pattern replacement.

- Object Model
  - Object keys — In many systems, objects need to have unique identifiers identifying them. UUIDs are one way to achieve this, but they are not readable and they may change as objects are restored or dynamically built from XML or database specifications. GRTL provides a flexible mechanism to implement a primary key for classes. For some classes the key may be the object UUID, while for others it may be the DCS tag or a network device identity. GRTL implements APIs to support custom object keys.
  - Object properties — Objects contain private and public parameters, states, and metrics. In many cases, you need to capture a richer object model for these parameters, states, and metrics than what is possible in G2. For example, you might want to capture if changes are allowed, the minimum value, the maximum value, the list of possible discrete values, the prompt label, units, or richer data types such as ISO date time, URL, color, or currency, for example. You might also want to capture such information at a central point and distribute it as needed to avoid inconsistencies and change management issues. Capturing such information is not possible at the G2 level. GRTL implements functionality to capture and query such information for properties.
  - Event model — To be able to loosely integrate modules or build distributed applications, a typical architecture is to use an event notification model to inform components at run time of state changes. COM, Java, and .NET support such event models. This approach avoids complex module hierarchies and dependencies including circular module

hierarchy requirements. G2 rules offer one form of event detection but typically are used at the class level instead of for specific instances, and they cannot be used in distributed applications. The GRTL event model is based on the G2 publish/subscribe model with the additional benefit that listeners can register at with classes to be notified of events from any instance of that class or any subclasses.

- Predefined base classes — GRTL includes several base classes such as pre-defined classes for datapoints and classes that implement a text as the key for instances.

- Instance Repository

  GRTL includes classes and APIs to build a data warehouse of dynamically created objects and classes without requiring the user to manage them as it would otherwise typically be required with G2 applications. In many applications, objects are created dynamically at runtime, including via user interface interactions, while users do not really required to access those objects. Rather, users view menus, tree views, and dialogs for those objects. The GRTL object repository provides an infrastructure for automatically managing the storage of those dynamically created objects.

- User Interface

  GRTL includes a default user interface that be extended and is the basis for G2-based applications such as ReThink, e-SCOR, and Optegrity. This user interface can be disabled, if required, and includes functionality such as:

  - Default menu bar that dynamically enables and disables functionality based on the user modes: system-administrator, developer, and modelers.
  - Popup menus build dynamically based on the context and supporting cascading submenus.
  - Three toolbars for basic operations, layout operations, and Web browsing.
  - Project tree view to organize the content of the application and quickly access content, such as data points, process models, event detection diagrams, fault models, and message browsers.
  - Class hierarchy tree view for developers to quickly access classes and review the hierarchy.
  - Module hierarchy tree view for developers to review the hierarchy and quickly access the system tables for each module.
  - Native palettes for developers and modelers to model processes and graphical languages.
  - Native dialogs for ease-of-use to configure properties of objects. The content of some dialogs is build dynamically based on the user mode. For example, a dialog for a message queue includes fields to configure the logging of activities in system-administrator mode but not in other modes.

Loading GRTL

To use the GRTL module, you must load or merge one or more of the following modules, which are located in the *g2i\kbs* directory:

- *grtl.kb* — Core classes, methods, and procedures, and core GRTL module settings, including module settings used to configure the user interface and top-level module system tables.
- *grtl-utils.kb* — Core item methods, utility functions and procedures, basic GRTL module settings, and GFR text localization.
- *grtl-icons.kb* — Icons used in various *g2i* modules.

For a list of classes in the *grtl-utils.kb* module, see Appendix A, GRTL-UTILS Module.

The *g2i-ui-demo.kb* is located in the *g2i\examples* directory. On Windows, you can load the demo from the Start menu.

Utility Functions

GRTL provides numerous utility functions areas to improve productivity and build consistency in applications. The following sections summarize those utility functions.

General

The core runtime module of the G2 Developers' Utilities is GRTL, which includes a set of general utilities in these areas:

- Module settings — A grtl-module-settings class for module initialization.
- Classes and instances — APIs for managing classes and instances.
- Lists, vectors, and sequences — APIs to empty lists, vectors, and sequences, replace values or remove values from vectors and sequences, and sort sequences.
- Time and time span — APIs to format or decode date, time, and durations. It supports encoding and decoding date, time, and durations in ISO-8601 format and XML structures, which many business areas use. Also supports a timer capability.
- Math — APIs to calculate the change and rate of time series, and to calculate random values, based on several distribution functions.
- Text and symbol parsing and formatting — APIs to parse text, clean up text for example by removing carriage returns, stripping leading spaces, tokenizing a string, and creating symbols from text.

- Module management — APIs to manage G2 modules such as creating and renaming a module and updating dependent modules.
- Paths and files — APIs to format path names, and to get a default directory.
- Server status and performance meters — APIs to query, enable, and disable server performance meters.

System Configuration

The G2 Developers' Utilities should be started with a configuration file to specify non-default values for configuration parameters. Certain configuration parameters must have non-default values for certain utilities to execute properly, so you should always run with a configuration file. The GRTL module provides APIs to query the configuration file.

The configuration file can be named anything and can be located anywhere. The default configuration file is called *config.txt*. The name and path of the configuration file is specified with the following command-line argument to the G2 launch command: *-configfile path*. You can use a pathname relative to the default directory.

For example:

```
g2.exe -ok g2.ok -kb kbs\g2i-dev.kb
-configfile Sites\Site1\config.txt
```

Configuration parameters are organized in the configuration file by subject area such as G2 modules or message queue, for example. Empty lines are permitted and ignored. Lines starting with a sharp sign (#) are comments, and are also permitted and ignored. For each group supporting configuration parameters, if a non-default value is desired for a parameter, then you must add a section to the file of the form:

```
[GROUP-NAME]

PARAMETER-NAME=parameter-value

PARAMETER-NAME=parameter-value
```

Group names and parameter names must be in upper case letter. The syntax of the parameter value is defined on a case-by-case basis. You can use forward slashes in path names for both UNIX and Windows. The content of the configuration file is typically imported into the G2 application upon startup time to provide initial configuration.

For example:

```
[GDSF]

RUN-MODE=online

ARCHIVING-MODE=file-archiving
```

ARCHIVING-DIRECTORY=d:/G2i/Sites/SimpleSite/Archives

RELATE-ALL-AT-START=no

AUTOMATIC-RELATES=no

Path names in the configuration file can start with *$INSTALLATION-DIRECTORY* or *$APPLICATION-ROOT-DIRECTORY* to refer to the bundle installation directory or to the application target directory.

In addition to relying on an external configuration file, GRTL as well as other G2 modules define a module setting class. For GRTL the module settings class is called grtl-module-settings. Because this class is a subclass of gfr-module-setting, the instance in the highest module within the module hierarchy is active. This enables other modules to provide custom configurations and overwrite the default configuration provided by GRTL. In addition, many attributes of the active grtl-module-setting instance are configured based on the content of the configuration file.

The combination of these two system configuration techniques provides the ability to configure applications by using instances of module settings, such as grtl-module-settings, which are typically specific to an application type (ReThink, Optegrity, etc.). They do not change for each site installation and deployment specific configuration for a specific application via the configuration file.

Localization

GRTL uses resource localization to support internalization and to make it easy to separate message formatting and code. The localization relies on the GFR localization capability; however, to ease the configuration, a global resource named grtl-text-localization-resource and defined in GRTL is used. By default, a local text resource is specified for the English language. If desired, you can create your own for other languages. GRTL defines wrapper APIs to access resources. These APIs should be used rather than the GFR APIs to enable improved implementations in the future.

Also upon startup the localization specification is loaded from one or more files and merged into a single GFR resource localization object defined in GRTL. These resource files to load and merge are defined in the system configuration file. You can specify the GFR resource files in the *config.txt* file, using a comma-separated list for the *APPLICATION-LOCALIZATION-FILES* resource. The files are be merged into the grtl-text-localization-resource object. The * character in the file name is replaced by the different supported languages to build the full filename. In addition, the filenames may start with *$INSTALLATION-DIRECTORY* or *$APPLICATION-ROOT-DIRECTORY* to refer the bundle installation directory or application target directory, which is resolved dynamically at runtime.

To add support for a new language, you create new instances of a gfr-local-text-resource, set the language, and set the gfr-resource-group to grtl-text-localization-resource. You also create an external text resource file containing the text for your

custom language and name it *resource-language.txt*. At startup, GRTL loads the external text resource files into the GFR text resource objects.

To avoid name clashes in the file and to make it easier to identify the purpose of each entry in the resource file, you need to use a convention. The symbolic key of each entry is composed of multiple name segments separated by a dot. For example, entries to specify configurations for menus are entered with the menu prefix following by the menu name (menu key) and followed by a keyword that identifies the subsection such as the label, the help string and the mnemonic for example. Resource specifications for dialogs will start with the DIALOG keyword followed by the name (key name) of the dialog and then followed by the specific subsection identifiers.

Below are examples of keywords that illustrate how to define resource specifications:

- MENU.GRTL-REPORTS.LABEL, "Reports"
- MENU.GRTL-REPORTS.MNEMONIC, "R"
- MENU.GRTL-REPORTS.ACCELERATOR, ""
- MENU.GRTL-REPORTS.TOOLTIP, "Reports"
- DIALOG.TRACK-ITEM.GET-PERIOD, "Enter the item tracking period:"
- DIALOG.TRACK-ITEM.NOTIFICATION, "Tracking item creation for the next [delay as a interval]"
- DIALOG.TRACK-ITEM.CAPTION, "Item Tracking"
- DIALOG.TRACK-ITEM.DONE, "Done tracking item creation. Use the Inspect command 'show on a workspace every item in the tracked item set' to view the tracked items."

Alternatively the resource file can also be specified in the command line with option *-resourcefile* path. The path argument should specify the directory and filename of the resources. In the following example, the asterisk (*) in the path is replaced with the target language to initialize the different gfr-local-text-resource instances assigned to the grtl-text-localization-resource group. You can have multiple resource files for each language in the same directory, and specify the directory and file pattern via the command line. For example:

```
g2.exe -ok g2.ok -kb kbs\g2i-dev.kb
-resourcefile Sites\Site1\resources-*.txt
```

Command-Line Options

GRTL includes APIs to query custom command-line arguments. The custom arguments are added to the command line that starts the G2 process. G2 applications can query these optional arguments.

User Interface

In addition to providing general utility functions and a basic object model, GRTL includes a basic user interface framework, which contains many elements typical in applications, including a menu bar, toolbars, application tree view navigation, palettes, and message browser. The framework is extensible and provides a rich starting point for building your own custom user interface. The framework also provides a consistent way to organize information and navigate applications. The user interface is the basis for several Gensym application products, including ReThink, e-SCOR, and Optegrity. When merging the different applications into G2, the user interface is automatically enhanced with the additional functionality. For example, nodes are added to the Project navigation tree view, menus are added to the menu bar submenus, and toolbars are added.

Prior to describing the user interface components, classes and APIs in subsequent chapters, the following paragraphs describe basic concepts used in the GRTL to build and manage the user interface. The *g2i-ui-demo.kb* provides very simple examples on how to use these APIs. On Windows, you can access the demo by choosing Start > Programs > Gensym G2 2011 > Examples > G2 Developers' Utilities > User Interface Demo.

User Modes

Depending on the user mode, menu entries may be included or excluded, and dialogs may include or exclude fields. Therefore, it is important to understand the built-in user modes. You can also define your own user modes.

Administrator

The G2 administrator should only be used to perform actions that cannot be done in any other mode. Otherwise, users should use one of the other modes.

System-Administrator

A user who is in the system-administrator mode has privileges to configure the server and perform other maintenance tasks. This includes configuring network connections, configuring log files, and adding or removing users. A user in system-administrator mode includes all functionality available in developer mode.

Developer

Users performing any customization requiring writing G2 code including creating items or classes should use developer mode. Menus, toolbars, and dialogs include additional fields in developer mode. For example, entries in the menu bar include tools to profile the application, and item properties dialogs may display additional attributes to specify custom procedures, for example. A user in developer mode includes all functionality available in modeler mode.

Modeler

Users configuring the application, such as creating or configuring domain objects, domain maps, event detection logic, fault models, or procedural flows should use modeler mode.

Operator

Users monitoring the application, such as managing messages and alarms should use operator mode. The operator mode user interface is much simpler than the other modes and focuses the operator on the core tasks of managing alarms and monitoring system performance.

User Preferences

While user modes already provide a large grouping of users per roles, you can also define user-specific preferences and permissions. For example, a user may specify the email address to forward messages when not logged in or have permission to delete messages. Permissions are typically configured by the system-administrator while the user can change his own user preferences. Both configurations are stored in user preference objects. GRTL keeps track of one user preference per user account.

The root class for user preferences is grtl-user-preferences. Other modules can define subclasses to add additional information. The active user preference class is defined in the active grtl-module-settings object.

User Sessions

When a user connects to a server via Telewindows, an instance of a grtl-user-session is created for each connection and associated to the g2-window. Instances of grtl-user-session contain runtime information about the user interface, such as menu or window handles. The content of grtl-user-session objects is private to GRTL and is not described further in this documentation.

Using a Secure G2

User preferences can be used in combination with a secure G2 server. User accounts available in a secure G2 are managed through the user preferences, using the same policies as described earlier. User accounts, including passwords, should be added to the *g2.ok* file as is usually done in G2.

In system-administrator mode, the user interface provides support for configuring users via dialogs. These menus are located in the Tools menu and are only visible in system-administrator mode.

Object Model

GRTL defines an object model with classes and related APIs that provide:

- An object-oriented environment composed of properties, methods, and events.
- A property model that enables you to define properties of objects beyond what G2 offers, including visual and constraint information such as the label, description, minimum value, maximum value, data type, whether the value is editable or browsable, and the category.
- Methods for getting and setting property values, using a generic API that takes the target property name as an argument. These APIs can be called remotely from another G2 applications or from a remote processes such as Visual Basic, .NET, XML, or Java, thereby simplifying integration with other applications.
- An event model that notifies listeners of changes that have or will occur in an object.
- A mechanism for referring to objects by keys. Keys should be unique for a class of objects; however, specific classes can implement keys, using different mechanism such as UUIDs or key attributes.

Object APIs

GRTL includes a set of APIs that are defined at the G2 item level to ensure the availability of basic functionality, such as creating items, deleting items with removal of related operator messages, constructors, destructors, and serialization and de-serialization of objects to structures, XML and text.

Object Keys

GRTL provides a universal naming system that allows different classes to use different attributes as a key name while abstracted from the application logic via generic APIs. Keys are human-readable IDs and must be unique for a class. Keys allow you to identify objects by unique ID that is application specific, such as a network-managed domain object OID or a DCS tag, and query objects based on these keys. This approach is different from the G2 UUID approach, because the key is not lost as items are transformed into another data type or re-created when retrieved from a file or database.

Advantages of using keys include:

- Keys are human readable.
- Keys can be defined and set at the application level.
- A preexisting class can support the key API without modification, provided it has an existing attribute to serve the purpose. The default key is the G2 UUID.

GRTL provides APIs to set the key, get the key, and query items given a class and a key.

Object Properties

Properties are public attributes that may or may not map directly to G2 attributes. For example, they may be virtual attributes in a sense that they are computed when queried or retrieved from related data structures. An example of a virtual attribute for operator messages is the name of the target or initiating item. At design time you can specify extended property information such as descriptions, units, minimum value, maximum value, list of discrete values, units, data type of property, such as ISO datetime, ISO duration, file, url, color, and timestamp, beyond the G2 data type. GRTL provides APIs to get and set property values.

The property information can either be specified by overloading the basic APIs or for ease of use specified statically on the class definition for definitions that are an instance of grtl-class-definition. To maintain an easy upgrade path for existing class definitions, GRTL also supports creating instances of grtl-attribute-property-details related to a class definition, which store the extended properties statically.

Property Types

The following table summarizes the property types, and the selected G2 data type to implement it:

Property Type	G2 type or class	Description
Truth-value	truth-value	A truth-value or boolean
Text	text	Any text
Symbol	symbol	Any symbol
Integer	integer	An integer
Float	float	A float
Quantity	quantity	A quantity
Currency	float	Any currency. The units property will determine the units of the currency.

Property Type	G2 type or class	Description
Datetime	quantity or no data type	G2 date-time format, which is a number relative to the start of the G2 process. It is converted to date time format as a string for sending the value.
Duration	quantity or no data type	G2 duration format, which is a number that can be specified in G2 format, for example, 3 weeks, 3 days and 2 hours.
ISO-time	integer	Time in ISO 8601 time format, for example, 15:45:00.000. The G2 data type is expressed in number seconds within a day. It ignores a subsecond resolution.
ISO-datetime	quantity or no data type	Date-time in ISO 8601 datetime format, for example, 1998-07-12:16:30:00.000. The G2 data type is expressed as quantity in G2 date time format.
ISO-duration	quantity or no data type	Duration in ISO 8601 duration format, for example, P30D. The G2 data type is expressed as a quantity in units of seconds.
ISO-date	quantity or no data type	Date in ISO 8601 date format, for example, 05-07-1995. In G2 it does not use any data type, such as the G2 data time format, but it ignores the time, which is set to midnight.
ISO-month	Integer	Month in ISO 8601 Gregorian month format, for example, --07--.
ISO-year	Integer	Year in ISO 8601 Gregorian year format, for example, 1998, 2002.
ISO-yearmonth	structure	Year and month in ISO 8601 Gregorian year month format, for example, 1998-07. The G2 structure has a month and year attribute.
ISO-Day	Integer	Day in ISO 8601 Gregorian day format, for example, ---12.

Property Type	G2 type or class	Description
ISO-monthday	structure	Month and day in ISO 8601 Gregorian month day format, for example, --07-12. The G2 structure has a month and day attribute.
Color	symbol	G2 color name.
File	text	A string that refers to a relative or the full pathname of a file.
Html	text	A string that contains HTML code.
Uri or url	text	A string that refers to a valid Uniform Resource Identifier.
Simple-logical-datapoint	grtl-simple-logical-datapoint	A simple logical datapoint that supports event notification.
Simple-text-datapoint	grtl-simple-text-datapoint	A simple text datapoint that supports event notification.
Simple-symbolic-datapoint	grtl-simple-symbolic-datapoint	A simple symbolic datapoint that supports event notification.
Simple-quantity-datapoint	grtl-simple-quantity-datapoint	A simple quantity datapoint that supports event notification.
Simple-integer-datapoint	grtl-simple-integer-datapoint	A simple integer datapoint that supports event notification.
Simple-float-datapoint	grtl-simple-float-datapoint	A simple float datapoint that supports event notification.
Derived-logical-datapoint	grtl-derived-logical-datapoint	A derived logical datapoint based on a formula that supports event notification.
Derived-text-datapoint	grtl-derived-text-datapoint	A derived text datapoint based on a formula that supports event notification.
Derived-Symbolic-datapoint	grtl-derived-symbolic-datapoint	A derived symbolic datapoint based on a formula that supports event notification.

Property Type	G2 type or class	Description
Derived-quantity-datapoint	grtl-derived-quantity-datapoint	A derived quantity datapoint based on a formula that supports event notification.
Derived-integer-datapoint	grtl-derived-integer-datapoint	A derived integer datapoint based on a formula that supports event notification.
Derived-float-datapoint	grtl-derived-float-datapoint	A derived float datapoint based on a formula that supports event notification.
External-logical-datapoint	grtl-external-logical-datapoint	An external logical datapoint that supports event notification.
External-text-datapoint	grtl-external-text-datapoint	An external text datapoint that supports event notification.
External-symbolic-datapoint	grtl-external-symbolic-datapoint	An external symbolic datapoint that supports event notification.
External-quantity-datapoint	grtl-external-quantity-datapoint	An external quantity datapoint that supports event notification.
External-integer-datapoint	grtl-external-integer-datapoint	An external integer datapoint that supports event notification.
External-float-datapoint	grtl-external-float-datapoint	An external float datapoint that supports event notification.

Property Constraints and Information

The following table summarizes the property information the server maintains and that you can query by calling grtl-get-property-type-info. It lists the attributes of structure returned by grtl-get-property-type-info that describe and constrain it. The structure can be generated dynamically, or statistically stored on grtl-class-definition items or in a grtl-attribute-property-details related to a class definition. In all cases, use grtl-get-property-type-info for queries.

Name	Valid Values	Description
property-name	Any symbol	Property name, which may correspond a G2 attribute name or to a virtual attribute name.
type	Any symbol	Property type. See previous table for valid values.
public	True or false	True if property is public; otherwise, it is considered private and should be accessed from other modules or UI components.
read-only	True or false	True if property is read only and cannot be set by users.
category	Any text	Category of property. Used by property grid view to group properties.
min	Any quantity	Minimum value of property for quantity types or the minimum text length for text attributes when edited by users in UI controls.
max	Any quantity	Maximum value of property for quantity types.
precision	Integer	Number of decimal values to display. Specifying 0 does not constraint it.
choices	Sequence of values	A fixed list of valid discrete values as a sequence or a procedure name as a symbol. The signature of the procedure specified in choices is: my-choices-proc-name (*itm*: class item {or any class}, *win*: class ui-client-item) -> *values*: sequence {list of text with the list of choices}
value	Value	A special property information keyword that can be passed to the getter/setter of datapoint objects to get or set the value of the datapoint.

initial-value	Value	A special property information keyword that can be passed to the getter/setter of datapoint objects to get or set the initial value of the datapoint.
history	Sequence	A special property information keyword that can be passed to the getter/setter of datapoint objects to get or set the history of the datapoint.
units	Any text	A special property information keyword that can be passed to the getter/setter of datapoint objects to get or set the units of the datapoint.
formula	Any text	A special property information keyword that can be passed to the getter/setter of datapoint objects to get or set the formula of a derived datapoint.

Object Event Model

Objects that are subclasses of grtl-event-source support an event model where notification events are sent to listeners. To register with an event source, a listener calls grtl-add-event-listener, grtl-add-attribute-listener, or grtl-add-datapoint-change-event-listener. The class definition to define event source objects needs to be an instance of grtl-event-source-class-definition rather than the basic G2 class-definition or object-definition. Instances need to inherit from grtl-event-source in order to support the event notification functionality.

The event model supports listeners running in the same G2 process or listeners running in a remote process on the same or on a different computer. The remote process can use G2 Gateway, G2 JavaLink, G2 ActiveXLink, or G2-to-G2 communication to register as a listener.

If a listener registers to a class, it will receive any event notification published by any instances of the registered class or by any subclass of the registered class. To process event notification, a listener must implement the grtl-event-notification method, which is the receiving method on any listener for all events. The *event-type* argument specifies the type or class of events, while *event-args* includes additional information for each event. Since events can be dispatched to remote processes, no reference to objects or items should be used in *event-args*. Instead, use the item key, as described above.

Events are dispatched by the event source by calling grtl-dispatch-event. The following table summarizes the events that are currently supported for various modules:

Event Type	Event Arguments (Examples)	Applicable to Instances of Class	Description
grtl-created	structure (EVENT-TYPE: the symbol grtl-created)	grtl-event-source	Sent when item is created by basic constructor if grtl-create is used to create an instance.
grtl-cloned	structure (EVENT-TYPE: the symbol grtl-deletion, SOURCE-ITEM-KEY: "key-of-item-that-was-cloned")	grtl-event-source	Sent when item is cloned by basic constructor if grtl-clone is used to clone an item.
grtl-deletion	structure (EVENT-TYPE: the symbol grtl-deletion)	grtl-event-source	Sent before item is deleted by basic destructor if grtl-delete is called to delete items.
grtl-key-changed	structure (EVENT-TYPE: the symbol key-changed, OLD-KEY: "previous-key")	grtl-event-source	Sent when the key property changed value.
grtl-properties-changed	structure (EVENT-TYPE: the symbol grtl-properties-changed, CHANGED-PROPERTIES: sequence (the symbol temperature, the symbol label))	grtl-event-source	Sent when the properties changed values.

grtl-datapoint-value-changed	structure (EVENT-TYPE: the symbol grtl-datapoint-value-changed, DATAPOINT-VALUE:12.2, DATAPOINT-COLLECTION-TIME: 340, DATAPOINT-KEY: "", DATAPOINT-STATUS: the symbol ok)	grtl-datapoint	Sent when the value of a variable-or-parameter changed.
grtl-datapoint-status-changed	structure (EVENT-TYPE: the symbol grtl-datapoint-status-changed, DATAPOINT-STATUS: the symbol ok)	grtl-datapoint	Sent when the value of a variable-or-parameter changed.
grtl-scheme-activated	structure (EVENT-TYPE: the symbol grtl-scheme-activated)	grtl-scheme	Sent when a scheme is activated.
grtl-scheme-deactivated	structure (EVENT-TYPE: the symbol grtl-scheme-deactivated	grtl-scheme	Sent when a scheme is deactivated.
grtl-scheme-value-changed	structure (EVENT-TYPE: the symbol grtl-scheme-value-changed, SCHEME-VALUE: 12.34)	grtl-scheme	Sent when a scheme's value changed.
gevm-queue-entry-attribute-change	structure (EVENT-TYPE: the symbol gevm-queue-entry-attribute-change, GEVM-EVENT-KEY: key, GEVM-EVENT-ATTRIBUTE: attribute)	gevm-gqs-queue	Sent when the attribute of an entry has changed.
gevm-acknowledged-event	structure (EVENT-TYPE: the symbol gevm-acknowledged-event, USER-NAME: "user-name")	gevm-event	Sent when an event has been acknowledged.

Instance Repository

GRTL uses instance repositories, also called data warehouses, to store user-defined instances that are created dynamically and are specific to a project. A repository creates instances of grtl-class-instance-holder on its subworkspace and stores the user-defined instances on the subworkspace of a grtl-class-instance-holder. This technique organizes instances by their class within an instance repository.

GRTL provides APIs to store instances and query them, as well as to locate the repository. If desired, you can create repositories dynamically. If the repository is created transparently, it is stored in the bin of a G2 module.

The G2 Developers' Utilities modules might create G2 objects dynamically without explicit clone and drop from the users. For example, grtl-user-preferences might be created dynamically or items might be created when the user clicks the New Instance menu choice in the tree view. By default, dynamically created objects are automatically stored in a repository organized by class in the top-level module. The repository is stored in the public bin of the top-level module, by default.

The repository hierarchy is as follows:

- The top-level repository is the Public Bin workspace in the top-level module, by default.
- The next level down in the hierarchy is the grtl-repository instance on the Public Bin workspace, typically the one with a key of "Default repository".
- The subworkspace of grtl-repository contains instances of grtl-class-instance-holder, which specify the class name of instances allowed on its subworkspace.
- The subworkspace of each grtl-class-instance-holder contains the dynamically created instances.

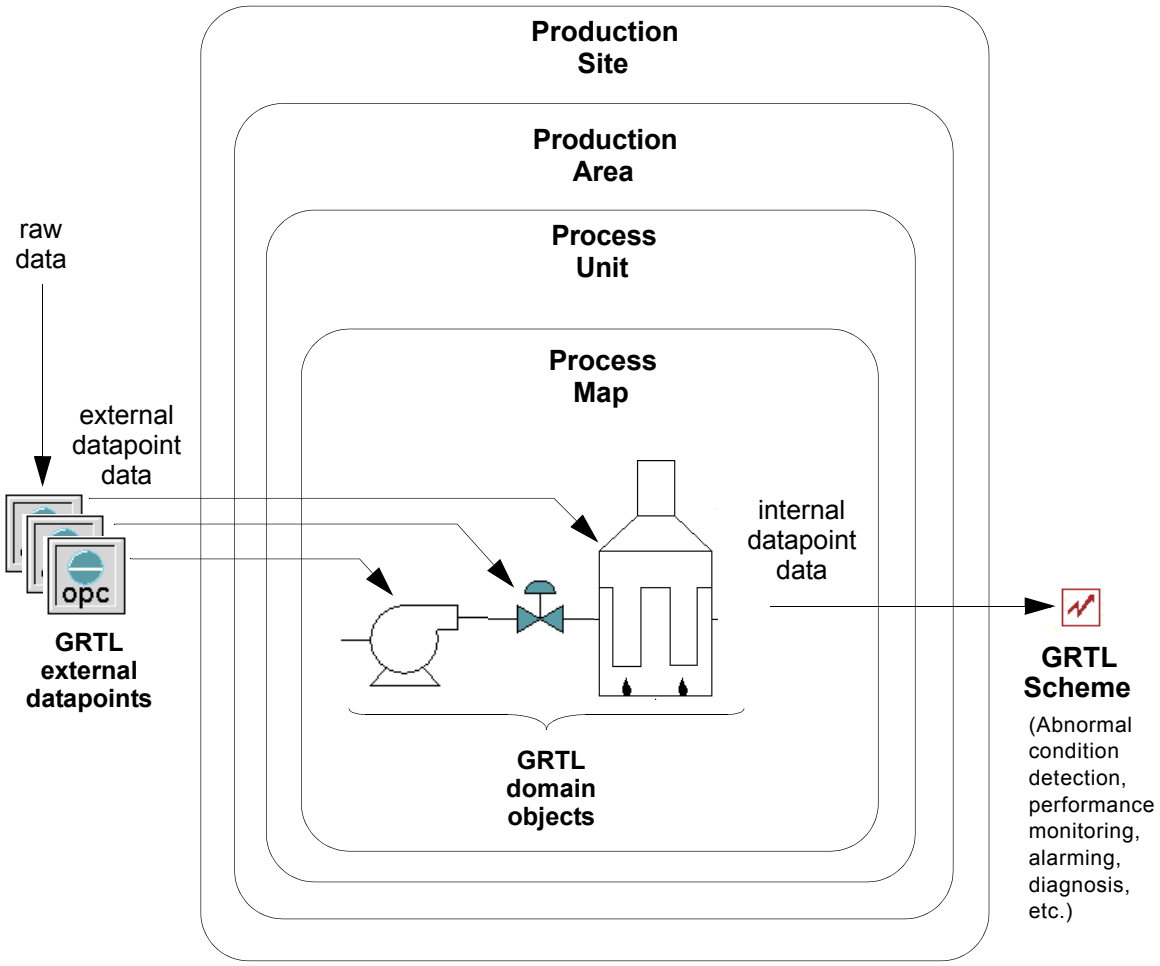
Datapoints, Schemes, Domain Objects, and Process Maps

GRTL provides an object model for datapoints, metrics, and schemes. Schemes implement any logic used to validate datapoints, monitor situations, detect events, perform diagnostics, or perform actions. You can implement schemes as G2 procedures or using a graphical language, such as G2 Event and Data Processing (GEDP) diagrams. Datapoints can trigger one or multiple schemes in three ways: enable a scheme, disable a scheme, or evaluate a scheme.

GRTL also provides an object model for process maps and domain objects. Process maps represent manufacturing processes, IT network topology, supply

chain networks, or any other type of process. They provide hierarchical containers in which to store domain objects, which define internal datapoints that obtain their data from external or derived datapoints. Domain objects provide graphical representations of elements in a process, such as process equipment or network elements. Domain objects can be connected to other domain objects, and they can be stored at various levels in the process map hierarchy. This means you can implement logic that reasons over connectedness and containment relationships, for example, for event detection and diagnostic reasoning. Domain objects are event sources with keys.

This figure shows the hierarchical containment of a manufacturing process that consists of several GRTL domain objects in a process map. The domain objects obtain their data from GRTL external datapoints and send internal datapoint data to a GRTL scheme:



Summary of Core Classes

This section briefly describes the core GRTL classes:

Class Name	Description
grtl-class-definition	A class definition supporting storing the name of the attribute used as a key on instances and the information on properties such as description, minimum value, and so on.
grtl-item	The root GRTL item.
grtl-item-with-key	An item that supports the definition of a user-defined key.
grtl-object	The root GRTL object.
grtl-object-with-key	An object that supports the definition of user-defined key.
grtl-workspace	The root class for GRTL workspaces.
grtl-repository	A repository or data warehouse for dynamically created instances that need to be stored by don't belong naturally to any other structure or workspace.
grtl-simple-logical-datapoint	A simple logical datapoint that supports event notification.
grtl-simple-text-datapoint	A simple text datapoint that supports event notification.
grtl-simple-symbolic-datapoint	A simple symbolic datapoint that supports event notification.
grtl-simple-quantity-datapoint	A simple quantitative datapoint that supports event notification.
grtl-simple-integer-datapoint	A simple integer datapoint that supports event notification.
grtl-simple-float-datapoint	A simple float datapoint that supports event notification.
grtl-derived-logical-datapoint	A derived logical datapoint, based on a formula that supports event notification.
grtl-derived-text-datapoint	A derived text datapoint, based on a formula that supports event notification.

grtl-derived-symbolic-datapoint	A derived symbolic datapoint, based on a formula that supports event notification.
grtl-derived-quantity-datapoint	A derived quantitative datapoint, based on a formula that supports event notification.
grtl-derived-integer-datapoint	A derived integer datapoint, based on a formula that supports event notification.
grtl-derived-float-datapoint	A derived float datapoint, based on a formula that supports event notification.
grtl-external-logical-datapoint	An external logical datapoint that is a subclass of gsi-data-service and supports event notification.
grtl-external-text-datapoint	An external text datapoint that is a subclass of gsi-data-service and supports event notification.
grtl-external-symbolic-datapoint	An external symbolic datapoint that is a subclass of gsi-data-service and supports event notification.
grtl-external-quantity-datapoint	An external quantitative datapoint that is a subclass of gsi-data-service and supports event notification.
grtl-external-integer-datapoint	An external integer datapoint that is a subclass of gsi-data-service and supports event notification.
grtl-external-float-datapoint	An external float datapoint that is a subclass of gsi-data-service and supports event notification.
grtl-ui-display	An event listener for datapoints to display the value of the datapoint.
grtl-process-value-display	An event listener for datapoints to display the value of the datapoint.

2

Module Settings

Describes module settings to configure operations of GRTL procedures and other modules relying on GRTL.

Introduction

The grtl-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.

grtl-module-settings

A GFR module settings that specifies GRTL-specific settings, such as configuration file, resource file, operator message procedures, default message properties, date and time formats and delimiters, default repository, indicator arrow behavior, default user preference, and performance meter properties.

Class Inheritance Path

grtl-module-settings, gfr-module-setting, object, item

Attributes

Attribute	Description
config-file-in-use	Specifies if the configuration file, *config.txt*, by default, is in use or not. The pathname to the configuration file can be specified as an option to G2 upon startup.
Allowable values:	Any truth-value
Default value:	false
config-file	The configuration file to use, which is specified at G2 startup by using the *-configfile* command-line option.
Allowable values:	Any text
Default value:	""
localization-file	The pathname to the GRTL resource file specified as a command-line option to G2 at startup. The pathname may contain an asterisk (*) in the file name, which is replaced with the language of the resource file to load, for example, english. The resource file is specified by using *-resourcefile* command-line option. The initial value of this configuration is specified in the *config.txt* file, which is read in at startup time.
Allowable values:	Any text
Default value:	""
Notes:	See Configuration File.

Attribute	Description
post-operator-message-procedure	The procedure called to post operator messages. For example, GEVM registers its own procedure to reroute the requests to GEVM.
Allowable values:	Any symbol
Default value:	_GRTL-POST-MESSAGE-TO-COMMON-MESSAGE-BOARD
delete-operator-message-procedure	The procedure called to delete operator messages. For example, GEVM registers its own procedure to reroute the requests to GEVM.
Allowable values:	Any symbol
Default value:	NONE
default-message-priority	The default priority of messages. For details, see the *G2 Event Manager User's Guide*.
Allowable values:	Any integer
Default value:	9
default-message-life-time	The default life time of messages. For details, see the *G2 Event Manager User's Guide*.
Allowable values:	Any integer
Default value:	-1
date-time-format	The default date and time format to use when formatting timestamps.
Allowable values:	The symbols: year-month-day-hour-mm-ss month-day-year-hour-mm-ss day-month-year-hour-mm-ss isox-date-time hour-mm-ss hour-mm:hh.hh hh.hh, dd.dd g2-time-stamp
Default value:	month-day-year-hour-mm-ss
Notes:	See Configuration File.

Attribute	Description
date-time-format-date-delimiter	The date delimiter when formatting timestamps to text.
Allowable values:	Any text
Default value:	"/"
Notes:	See Configuration File.
date-time-format-time-delimiter	The time delimiter when formatting timestamps to text.
Allowable values:	Any text
Default value:	":"
Notes:	See Configuration File.
date-time-format-date-time-delimiter	The separator between the time and date when formatting timestamps to text.
Allowable values:	Any text
Default value:	" "
utc-offset	An integer giving the time offset for the current time zone from the UTC time, for example, -5 for the East Coast of the US. This attribute is currently used to generate the time axis for charts.
Allowable values:	integer
Default value:	0
Notes:	See Configuration File.
default-repository-module	The G2 module name to use by default when storing items in repositories. For details, see the *G2 Event Manager User's Guide*.
Allowable values:	Any symbol
Default value:	GRTL
Notes:	See Configuration File.
indicator-arrow-hide-after-timeout	When set to true, automatically removes an indicator arrow after the specified timeout period, which you can display by using grtl-view-object.

Attribute	Description
Allowable values:	Any truth-value
Default value:	false
Notes:	See Configuration File.
indicator-arrow-default-timeout	The timeout period to wait before removing indicator arrows pointing to objects if indicator-arrow-hide-after-timeout is enabled.
Allowable values:	Positive integer
Default value:	60
Notes:	See Configuration File.
indicator-arrow-default-color	The default color of the indicator arrow pointing to items, which you can display by using grtl-view-object.
Allowable values:	G2 color name as a symbol.
Default value:	RED
Notes:	See Configuration File.
user-preferences-class-name	When GRTL detects a user connecting to the server, the user preference class to use for creating and managing user preferences, and for updating the user interface for different user modes.
Allowable values:	Any class name that is a subclass of grtl-user-preference.
Default value:	grtl-user-preferences
user-preferences-filename	The file name used to import and export the user configuration.
Allowable values:	text
Default value:	""
Notes:	See Importing and Exporting User Preferences.
available-user-modes	The list of available user modes.
Allowable values:	sequence

Attribute	Description
Default value:	(the symbol administrator, the symbol system-administrator, the symbol developer, the symbol modeler, the symbol operator)
performance-meter-enabled	If true, enables performance meters.
Allowable values:	Any truth-value
Default value:	false
performance-meter-lag-time	The lag time of performance meters.
Allowable values:	Positive integer representing the lag time in seconds.
Default value:	60
performance-meter-update-interval	The update interval of performance meters.
Allowable values:	Positive integer representing the duration in seconds.
Default value:	60
performance-meter-maximum-history-age	The maximum age of the history of performance meters.
Allowable values:	Positive integer or 0 to disable it
Default value:	0
performance-meter-nb-of-historical-values	The maximum number of points to keep in the history for performance meters.
Allowable values:	Positive integer or 0 to disable it.
Default value:	120
user-interface-monitoring-thread-refresh-period	The duration in seconds with which to refresh metrics displayed in UIL dialogs. This is only relevant for some dialogs, such as the ones in Optegrity.
Allowable values:	Any item or value
Default value:	30
Notes:	See Configuration File.

Attribute	Description
installation-directory	The default installation directory.
Allowable values:	text
Default value:	""
Notes:	See Configuration File.
application-name	Typically set by the gfr-startup-procedure to the application. For example, it is used to display the application name in the title bar of the Telewindows session.
Allowable values:	text
Default value:	""
application-home-directory	The directory where application-specific files should be stored, for example user-defined models. You typically set this as a configuration in the *config.txt* resource file. This enables you to specify a directory that is different from the installation directory. The pathname may start with the pattern $INSTALLATION-DIRECTORY, which is replaced at runtime by their appropriate values.
Allowable values:	text
Default value:	The current installation directory
Notes:	See Configuration File.
application-default-url	The default URL to access the application via the Web. The initial value of this configuration is specified in the *config.txt* file, which is read in at startup time.
Allowable values:	text
Default value:	none
Notes:	See Configuration File.

Attribute	Description
application-is-web-hosted	If true, specifies that the application is running in a hosted environment, which means, for example, it can restrict access to some functionality. The initial value of this configuration is specified in the *config.txt* file, which is read in at startup time.
Allowable values:	text
Default value:	false
Notes:	See Configuration File.
application-valid-window-ui-styles	A list of symbols specifying acceptable UI styles in Telewindows. The options include any of the possible values for the g2-window-ui-style hidden attribute of g2-window. When Telewindows is connected, if this attribute contains any values and if the g2-window-ui-style of the g2-window is not a member of the list, the connection is automatically closed.
Allowable values:	sequence
Default value:	sequence( )
enable-menus-and-toolbars-upon-startup	If true, displays the GRTL menu bar, toolbars, and popup menus. If false, does not activate and display them. If true, also updates the item configurations of the kb-configuration system table at startup to redirect keyboard and mouse actions, to display popup menus, or to restrict access to the KB, for example.
Allowable values:	truth-value
Default value:	false
enable-user-audit-file	If true, logs all user log ins, log outs, and change mode activities. The log file is specified in the user-audit-file. The initial value of this configuration is specified in the *config.txt* file, which is read in at startup time.
Allowable values:	truth-value

Attribute	Description
Default value:	false
Notes:	See Configuration File.
user-audit-file	The audit log file to use to log activities. The path name may start with the pattern $APPLICATION-ROOT-DIRECTORY or $INSTALLATION-DIRECTORY, which is replaced at runtime by their appropriate values. The initial value of this configuration is specified in the *config.txt* file, which is read in at startup time.
Allowable values:	text
Default value:	The current installation directory
Notes:	See Configuration File.
add-missing-message-localization-to-gfr-resource	When true and code attempts to fetch the resource value from the grtl-resource-group and the entry is missing, the key and default value are automatically added to the resource group. This feature enables you to quickly capture all resource keys and their default values simply by navigating through the user interface.
Allowable values:	truth-value
Default value:	true
minimum-scheduling-interval	The length of the G2 clock tick. For details, see System Tables in the *G2 Reference Manual*.
Allowable values:	a subsecond time interval or continuous
Default value:	2 ms
attribute-display-update-interval	Specifies the frequency with which to update attribute tables. For details, see System Tables in the *G2 Reference Manual*.
Allowable values:	continuous | float (between 0.0 seconds and 0.5 seconds)
Default value:	0.5 seconds

Attribute	Description
interface-mode	Specifies which interface mode G2 uses. For details, see System Tables in the *G2 Reference Manual*.
Allowable values:	{interruptible interface service | always service interface first}
Default value:	interruptible interface service
start-kb-after-load?	Controls whether G2 is started immediately after loading a KB. If this attribute is set to yes, G2 is started after loading the KB. For details, see System Tables in the *G2 Reference Manual*.
Allowable values:	{yes | no}
Default value:	no
connection-caching-enabled?	Determines whether graphical connections are cached. For details, see System Tables in the *G2 Reference Manual*.
Allowable values:	{yes | no}
Default value:	no
backward-compatibility-features	Lets you disregard certain changes made in recent G2 releases. For details, see System Tables in the *G2 Reference Manual*.
Allowable values:	{none | ignore duplicate list element error | extra vertices in g2-get-connection-vertices}
Default value:	none
kb-item-configuration	The KB-level configuration statements. For details, see System Tables in the *G2 Reference Manual*.
Allowable values:	For a complete description of using configuration statements, refer to Configurations.
Default value:	
kb-item-configuration-when-web-hosted	The KB-level configuration statements when application is web hosted.

Attribute	Description
Allowable values:	For a complete description of using configuration statements, refer to Configurations in the *G2 Reference Manual*.
Default value:	
kb-item-configuration-for-classes	Used to configure mouse and keyboard actions for classes, for example, to display GRTL popup menus using right click or displaying the properties dialog using double click. For details, see System Tables in the *G2 Reference Manual*.
Allowable values:	
Default value:	
kb-item-configuration-for-classes-enabled-for-user-modes	Enables item configuration for specific user modes.
Allowable values:	sequence
Default value:	sequence (the symbol system-administrator, the symbol developer, the symbol modeler)
kb-item-configuration-for-classes-enabled	Enables or disables item configurations for all classes.
Allowable values:	truth-value
Default value:	true
kb-main-menu-user-restrictions	Lets you restrict all menu choices on the Main Menu. The default is none. For details, see System Tables in the *G2 Reference Manual*.
Allowable values:	For a complete description of using configuration statements, refer to Configurations.
Default value:	none
kb-keyboard-command-restrictions	Lets you exclude or include global keyboard commands while in a user mode. For details, see System Tables in the *G2 Reference Manual*.

Attribute	Description
Allowable values:	For a complete description of using configuration statements, refer to Configurations.
Default value:	none
kb-initial-g2-user-mode-for-this-kb	Specifies a default user mode for the KB. The default is none, which means that the user is in administrator mode (the only system-defined user mode). For details, see System Tables in the *G2 Reference Manual*.
Allowable values:	For a complete description of user modes, refer to Configurations.
Default value:	none
kb-automatically-show-developer-menu-bar	Determines whether the developer menu bar appears in Telewindows. For details, see System Tables in the *G2 Reference Manual*.
Allowable values:	on pause, reset, or initial connection | on | never
Default value:	on pause, reset, or initial connection
show-selection-handles	If update-top-level-system-tables and enable-menus-and-toolbars-upon-startup of the active grtl-module-settings are both true, then show-selection-handles of the drawing-parameters system table will be set to the value of the show-selection-handles of the active grtl-module-settings.
Allowable values:	truth-value
Default value:	true
update-top-level-system-tables	When true, updates the top-level module's system tables.
Allowable values:	truth-value
Default value:	true

Configuration File

This table describes the settings in the configuration file (*config.txt*, by default), the associated group, and the attributes in the grtl-module-settings object that they configure at startup:

/

Group	Configuration File Settings	GRTL Module Settings Attributes/ Description
GRTL	*APPLICATION-LOCALIZATION-FILES=*	localization-file
GRTL	*DATE-TIME-FORMAT=month-day-year-hour-mm-ss*	date-time-format
GRTL	*DATE-TIME-FORMAT-DATE-DELIMITER=* "/"	date-time-format-date-delimiter
GRTL	*DATE-TIME-FORMAT-TIME-DELIMITER=* ":"	date-time-format-time-delimiter
GRTL	*UTC-OFFSET*	utc-offset
GRTL	*REPOSITORY-MODULE=top-level*	default-repository-module
GRTL	*INDICATOR-DELETE-BY-DEFAULT=true*	indicator-arrow-hide-after-timeout
GRTL	*INDICATOR-DEFAULT-TIMEOUT=60*	indicator-arrow-default-timeout
GRTL	*INDICATOR-DEFAULT-COLOR=red*	indicator-arrow-default-color
GRTL	*INDICATE-ITEMS=false*	indicate-items-upon-menu-selection of the active grtl-default-user-preference
GRTL	*USER-INTERFACE-REFRESH-PERIOD=5*	user-interface-monitoring-thread-refresh-period
GRTL	*APPLICATION-ROOT-DIRECTORY=$INSTALLATION-DIRECTORY*	application-home-directory
GRTL	*APPLICATION-URL=*	application-default-url
GRTL	*APPLICATION-IS-WEB-HOSTED=false*	application-is-web-hosted
GRTL	*USER-AUDIT-FILE-ENABLED=false*	enable-user-audit-file
GRTL	*USER-AUDIT-FILE=$APPLICATION-ROOT-DIRECTORY\logs\user-audit-trail.csv*	user-audit-file

Group	Configuration File Settings	GRTL Module Settings Attributes/ Description
GRTL	*APPLICATION-ERROR-INFORM-ENABLED=true*	When true, enables error logging to the G2 Logbook
GRTL	*APPLICATION-ERROR-LOG-ENABLE=true*	When true, enables error logging to the log file specified by APPLICATION-ERROR-LOG-FILE.
GRTL	*APPLICATION-ERROR-LOG-FILE=$APPLICATION-ROOT-DIRECTORY\logs\kb-errors.log*	Specifies the default location of the log file. The directory can refer to $INSTALLATION-DIRECTORY, which is the default installation directory.
GRTL	*USER-PREFERENCES-CONFIGURATION-FILE=$APPLICATION-ROOT-DIRECTORY/g2i/data/user-preferences.txt*	The location of the configuration file for all grtl-user-preferences. The file contains the preferences for all defined user preferences, each in its own section. The settings are imported from the file to configure the user preference objects in G2.

KB Initialization

Describes KB initialization procedures.

Introduction

GRTL provides a mechanism to configure and update the configuration of system tables of the top-level KB. System configurations are defined in the system tables of the top-level G2 KB module. The top-level KB module typically contains the data, domain objects, models, logic—such as event detection, test flow diagrams, workflows, or correlation specifications—of the application, which is required to maintain a consistent modularization of the G2 applications. However, required modules implementing the algorithm or the user interface may require certain configurations in the G2 system tables. In addition, as new versions of required modules are deployed, they may need to update the configuration of top-level modules.

GRTL provides functionality for configuring the KB, using a two-phased initialization:

- The first phase relies on the gfr-startup-object startup procedures.
- Once all instances of gfr-startup-object have been initialized, GRTL calls the grtl-setup-kb-configuration procedure, which calls every procedure instance of grtl-kb-configuration-procedure in the module hierarchy, starting from the lowest-level module and going to the highest-level module.

The GRTL GFR startup procedure in *grtl.kb* performs the following initializations:

```
conclude that the prefer-native-logbook of logbook-parameters = true;
conclude that the prefer-native-message-board of message-board-parameters = true;
```

```
conclude that the prefer-native-login-dialog of miscellaneous-parameters = true;
conclude that the show-uuids-in-attribute-tables of miscellaneous-parameters = true;
```

The signature of every grtl-kb-configuration-procedure must be:

```
my-kb-configuration-procedure
    (kb-config: class kb-configuration, timings: class timing-parameters,
    misc-parameters: class miscellaneous-parameters, win: class g2-window)
```

The grtl-kb-configuration-procedure performs the initialization based on the configurations in the grtl-module-settings.

If the update-top-level-system-tables attribute of the active grtl-module-settings is true, GRTL does the following:

- Sets the minimum-scheduling-interval of the timing-parameters system table to 2ms, by default. The specific setting is taken from the minimum-scheduling-interval attribute of the active grtl-module-settings instance.
- Sets the attribute-display-update-interval of the timing-parameters system table to 0.5 seconds, by default. The specific setting is taken from the attribute-display-update-interval attribute of the active grtl-module-settings instance.
- Sets the interface-mode of the timing-parameters system table to "always service interface first", by default. The specific setting is taken from the timing-parameters attribute of the active grtl-module-settings instance.
- Sets the show-uuids-in-attribute-tables of the miscellaneous-parameters system table to true, which is critical for GRTL to work correctly.
- Sets the connection-caching-enabled? of the miscellaneous-parameters system table to the specific setting from the connection-caching-enabled? attribute of the active grtl-module-setting instance.
- Sets the backward-compatibility-features of the miscellaneous-parameters system table to "print means print-to-server in configurations, inconsistent behavior of move in configurations, extra vertices in g2-get-connection-vertices", by default. The specific setting from the backward-compatibility-features attribute of the active grtl-module-setting instance.

If the enable-menus-and-toolbars-upon-startup and update-top-level-system-tables attributes of the active grtl-module-settings object are both true, the grtl-kb-configuration-procedure instance in GRTL does the following:

- Configures the instance-configuration of the kb-configuration system table so that:
  - Double-clicking or pressing the F4 key displays the properties dialog, and right clicking displays the item popup menu.
  - Pressing F2 on an item implies table in developer, system-administrator, or administrator mode.

  - Pressing F3 on an item implies hidden-table in developer, system-administrator, or administrator mode.
  - Pressing ALT + right click implies show-class-definition.
  - Pressing CTRL + right click on any object implies go-to-subworkspace.
  - Pressing CTRL + right click on any kb-workspace implies hide-workspace.
  - Pressing SHIFT + right click on any kb-workspace implies highlight-superior-item.
- Configures the item-configuration of the kb-configuration system table so that:
  - The item-configuration of kb-configuration is set to the kb-item-configuration attribute of the active grtl-module-settings.
  - If the application-is-web-hosted of grtl-settings is true, the content of the kb-item-configuration-when-web-hosted attribute of the active grtl-module-settings is appended to the item-configuration of kb-configuration.
  - The item-configuration of kb-configuration is appended with configurations to install the keyboard accelerators if any are defined and enabled.
  - If kb-item-configuration-for-classes-enabled of the active grtl-module-settings is true and if enable-menus-and-toolbars-upon-startup is also true, configurations are added to the item-configuration of the kb-configuration to enabled for the GRTL popup menus for specific classes. The specific configuration per class is defined in the kb-item-configuration-for-classes attribute of the active grtl-module-settings and for the user modes specified in the kb-item-configuration-for-classes-enabled-for-user-modes attribute of the active grtl-module-settings.
- Changes the text of the main-menu-user-restrictions of the kb-configuration system table to the following, by default, though the exact setting is taken from the kb-main-menu-user-restrictions attribute of the active grtl-module-settings instance.

  ```
  "unless in administrator, developer, or modeler mode: main menu choices
  exclude absolutely: change-mode;
  unless in administrator mode: main menu choices exclude absolutely:
  shut-down-g2, launch-online-help;
  when in operator or modeler mode: main menu choices exclude absolutely:
  new-workspace, get-workspace, inspect, system-tables, run-options,
  change-mode, miscellany, pause, reset, start, resume, restart, save-kb";
  ```

- Changes the text of the keyboard-command-restrictions of the kb-configuration system table to the following, by default, though the exact setting is taken

from the kb-keyboard-command-restrictions attribute of the active grtl-module-settings instance.

"unless in administrator or developer mode: global keyboard commands exclude absolutely: pause";

- Sets the initial-g2-user-mode-for-this-kb of the kb-configuration system table to "modeler", by default, though the exact setting is taken from the kb-initial-g2-user-mode-for-this-kb attribute of the active grtl-module-settings instance.
- Changes the automatically-show-developer-menu-bar of the menu-parameters system table to the kb-automatically-show-developer-menu-bar of the active grtl-module-settings instance.
- Changes the show-selection-handles of the drawing-parameters system table to the show-selection-handles of the active grtl-module-settings instance.

Instances of grtl-kb-configuration-procedure in modules higher in the module hierarchy may overwrite these settings.

To reconfigure the KB, applications may call the following APIs:

grtl-setup-kb-configuration
(*kb-config*: class kb-configuration, *timings*: class timing-parameters, *misc-parameters*: class miscellaneous-parameters, *win*: class g2-window)

Initializes the kb-configuration, timing-parameters, and miscellaneous-parameters system tables.

grtl-configure-kb-item-configuration-for-classes
(*kb-config*: class kb-configuration, *classes-to-configure*: sequence)

Configures the instance-configuration of the kb-configuration system table for the specified classes in *classes-to-configure*, which is a sequence of symbols, for example, to enable popup menus. The specific configuration per class is defined in the kb-item-configuration-for-classes attribute of the active grtl-module-settings and for the user modes specified in the kb-item-configuration-for-classes-enabled-for-user-modes attribute of the active grtl-module-settings.

Part II

Utilities

Chapter 4: Classes and Instances

Describes procedures to manage classes and instances.

Chapter 5: Lists and Arrays

Describes the permanent item list and array classes, and a method and procedures for manipulating lists.

Chapter 6: Vectors

Describes the vector class and methods for manipulating vectors.

Chapter 7: Sequences

Describes procedures to manipulate sequences.

Chapter 8: Timers

Describes a timer class and procedures to create timers and get the clock time.

Chapter 9: Time

Describes procedures and functions for converting numeric and textual representations of time and durations.

Chapter 10: Math

Describes procedures and functions for mathematical operations.

Chapter 11: Text Manipulation

Describes procedures and functions for manipulating text and converting text to symbols.

Chapter 12: KB Modules

Describes procedures for manipulating KB modules.

Chapter 13: Files and Directories

Describes procedures for manipulating file names and directory names.

Chapter 14: Server Status and Performance Meters

Describes built-in performance meters and procedures to query the status of the server and performance meters.

Chapter 15: Text Localization

Describes procedures for localizing text in different languages.

Chapter 16: Application Configuration

Describes procedures for retrieving application configuration information from a configuration file.

Chapter 17: Command Lines

Describes procedures for retrieving options added to the command line that started G2.

4

Classes and Instances

Describes procedures to manage classes and instances.

Introduction

GRTL provides a set of general class and instance operations for:

- Creating class definitions.
- Generating a list of subclasses
- Getting a list of attributes for a class.
- Getting the name or key of an item.

- Determining whether a class is a superior class of another class.
- Determining whether a relation is valid.

grtl-create-a-class-definition

Synopsis

grtl-create-a-class-definition
(*class-name*: symbol, *ancestry*: sequence, *workspace-or-repository*: item)

Argument	Description
class-name	The class to create.
ancestry	A sequence of symbols that are the class names making up the inheritance path. The first entry is the direct superior class. The next class is the superior class of that class, as so on. The number of ancestors provided is unlimited.
workspace-or-repository	A kb-workspace or grtl-repository instance where newly created classes are stored. If you specify any other value, the resulting class is still created, but it is not stored anywhere.

Description

Checks if a class exists and, if not, creates a class hierarchy, using the provided *class-name* and *ancestry*. Newly created classes are either stored on the workspace specified in *workspace-or-repository*, if it is a kb-workspace, or in the repository, if it is a grtl-repository. If classes are missing, it creates a grtl-missing-class-definition error.

grtl-generate-list-of-subclasses

Synopsis

grtl-generate-list-of-subclasses
(*class-name*: symbol, *list-of-classes*: sequence)
-> *class-names*: sequence

Argument	Description
class-name	The root class name.
list-of-classes	A sequence of symbols to extend with new class names.

Return Value	Description
class-names	The resulting list of class names as a sequence of symbols.

Description

Appends *class-name* and all instantiable subclasses to *list-of-classes*, then returns the resulting list as a sequence of symbols.

grtl-get-list-of-attributes-for-class-and-subclasses

Returns a list of symbols naming the attributes of a class and subclasses.

Synopsis

grtl-get-list-of-attributes-for-class-and-subclasses
(*class-or-classname*: item-or-value, *editable-attributes-only*: truth-value, *user-mode*: symbol, *attributes*: sequence, *attributes-to-exclude*: sequence)
-> *attributes*: sequence

Argument	Description
class-or-classname	The root class or class name whose attribute names to get.
editable-attributes-only	If true, return only editable attributes.
user-mode	A symbol naming a user mode. If specified, returns only attributes visible in the specified user mode.
attributes	If specified, appends the list of attributes to this list.
attributes-to-exclude	A sequence of attributes to exclude from the returned list.

Return Value	Description
attributes	A sequence of symbolic attribute names for the given class.

grtl-get-name-of-item

Synopsis

grtl-get-name-of-item
    (*itm*: item)
    -> *name*: text

Argument	Description
itm	Any item.

Return Value	Description
name	The name of the item.

Description

Returns the name of an item as a text. This API is similar to g2-name-for-item except that it always returns the result as a text.

grtl-get-name-or-key-of-item

Synopsis

grtl-get-name-or-key-of-item
    (*itm*: item)
    -> *name-or-key*: text

Argument	Description
itm	Any item.

Return Value	Description
name-or-key	The name of the item or its key.

Description

Returns the name of an item, if the name exist, as a text; otherwise, it returns the key of the grtl-item-with-key. See grtl-item-with-key.

grtl-get-name-or-key-of-top-level-item

Synopsis

grtl-get-name-or-key-of-top-level-item
    (*itm*: item)
    -> *name-or-key*: text

Argument	Description
itm	Any item.

Return Value	Description
name-or-key	The name of the item or its key.

Description

Searches for the top-level item of *itm* if the item is an embedded object of an attribute, and returns the name of the top-level item, if the name exist, as a text; otherwise, it returns the key of the top-level grtl-item-with-key. See grtl-item-with-key. The top-level item is the item that has no superior item, that is, is not defined as an embedded object in an attribute of a superior item.

grtl-get-top-level-item

Synopsis

grtl-get-top-level-item
    (*itm*: item)
    -> *item*: class item

Argument	Description
itm	Any item.

Return Value	Description
item	The top-level item.

Description

Returns the top-level item if the *itm* is an embedded object of an attribute. The top-level item is the item that has no superior item, that is, is not defined as an embedded object in an attribute of a superior item.

grtl-is-superclass

Synopsis

grtl-is-superclass
    (*superclass*: symbol, *subclass*: symbol)
    -> *return-value*: truth-value

Argument	Description
superclass	A symbol for the superior class.
subclass	A symbol for the subclass.

Return Value	Description
return-value	True if *subclass* is in the class inheritance path of *superclass*.

Description

Returns true if *subclass* is in the class inheritance path of *superclass*.

grtl-is-valid-relation

Synopsis

grtl-is-valid-relation
(*relation-name*: symbol, *class-def1*: class class-definition,
class-def2: class class-definition)
-> *return-value*: truth-value

Argument	Description
relation-name	The name of the relation or its inverse.
class-def1	A class definition.
class-def2	A class definition.

Return Value	Description
return-value	True if the relation exists between *class-def1* and *class-def2*.

Description

Verifies if there exists a relation *relation-name* between *class-def1* and *class-def2*. This procedure assumes that the relation is defined correctly. If the relation is valid, it returns true; otherwise, it returns false.

5

Lists and Arrays

Describes the permanent item list and array classes, and a method and procedures for manipulating lists.

Introduction

GRTL provides permanent list and array classes, and a method on g2-list to empty the list. It also provides a number of procedures for manipulating lists, including copying, merging, and sorting lists, converting a list to a sequence, deleting the contents of a list, and removing a subset of the list.

Classes

grtl-permanent-item-array

An item array whose array-is-permanent attribute is initially true.

Class Inheritance Path

grtl-permanent-item-array, item-array, g2-array, object, item

grtl-permanent-item-list

An item list whose list-is-permanent attribute is initially true.

Class Inheritance Path

grtl-permanent-item-list, item-list, g2-list, object, item

Methods and Procedures

Methods

g2-list::grtl-empty-list

Procedures

grtl-convert-list-to-sequence
grtl-copy-list
grtl-delete-contents
grtl-delete-list-and-contents
grtl-merge-lists
grtl-one-key-ascending-sort
grtl-one-key-descending-sort
grtl-purge-duplicates-from-list
grtl-quicksort-list
grtl-remove-subset-list

g2-list::grtl-empty-list

Synopsis

g2-list::grtl-empty-list
    (*list*: class g2-list)

Argument	Description
list	The g2-list to empty.

Description

Removes all elements from a list. If the list contains items, the items are not deleted. Use grtl-delete-contents to delete the items as well as removing them from the list.

grtl-convert-list-to-sequence

Synopsis

grtl-convert-list-to-sequence
    (*item-list*: class item-list)
    -> *seq*: sequence

Argument	Description
item-list	The item-list to convert.

Return Value	Description
seq	The sequence that contains the item-list elements.

Description

Converts an item-list to a sequence, preserving the contents of the original list.

grtl-copy-list

Synopsis

grtl-copy-list
    (*source-list*: class item-list, *destination-list*: class item-list)

Argument	Description
source-list	The item-list to copy elements from and to be emptied.
destination-list	The item-list into which to copy elements.

Description

Copies the contents of *source-list* to *destination-list*. It appends the contents of *source-list* to the end of *destination-list*. The source list remains unchanged.

grtl-delete-contents

Synopsis

grtl-delete-contents
    (*list-or-sequence*: item-or-value)

Argument	Description
list-or-sequence	A g2-list or a sequence.

Description

Deletes the elements from a list or sequence. If passed a list that contains items, this procedure also deletes the items.

grtl-delete-list-and-contents

Synopsis

grtl-delete-list-and-contents
    (*item-list*: class item-list)

Argument	Description
item-list	The item-list whose contents and instance to delete.

Description

Deletes every item in an item-list, then deletes the item-list instance. If the item-list to delete is permanent, this procedure makes the list transient before deleting it.

grtl-merge-lists

Synopsis

grtl-merge-lists
(*destination-list*: class item-list, *source-list*: class item-list)

Argument	Description
destination-list	The item-list into which to transfer elements.
source-list	The item-list from which to transfer elements.

Description

Transfers the content of *source-list* into *destination-list*, emptying the contents of *source-list* and merging it into *destination-list*. The items in *source-list* are appended to the end of *destination-list*.

grtl-one-key-ascending-sort

Synopsis

grtl-one-key-ascending-sort
    (*unsorted-item-list*: class item-list, *key*: symbol)

Argument	Description
unsorted-item-list	The unsorted item-list.
key	The name of an attribute of an item in the unsorted list upon which to base the sort order.

Description

Sorts the contents of *unsorted-item-list*, based on a *key*, which is an attribute whose value the procedure uses to sort the elements. The item-list is sorted from minimum to maximum, based on the values in the *key* attribute. Only items with an attribute named by the key are contained in the returned list. See also grtl-quicksort-list.

grtl-one-key-descending-sort

Synopsis

grtl-one-key-descending-sort
    (*unsorted-item-list*: class item-list, *key*: symbol)

Argument	Description
unsorted-item-list	The unsorted item-list.
key	The name of an attribute of an item in the unsorted list upon which to base the sort order.

Description

Sorts the contents of *unsorted-item-list*, based on a *key*, which is an attribute whose value the procedure uses to sort the elements. The item-list is sorted from maximum to minimum, based on the values in the *key* attribute. Only items with an attribute named by the key are contained in the returned list. See also grtl-quicksort-list.

grtl-purge-duplicates-from-list

Synopsis

grtl-purge-duplicates-from-list
    (*source-list*: item-list, *destination-list*: item-list)

Argument	Description
source-list	The item-list with duplicate elements.
destination-list	The item-list without duplicate elements.

Description

Copies the contents of *source-list* into *destination-list*, removing duplicates from *source-list*. If *destination-list* is initially empty, after this procedure returns, it contains one of each element in *source-list*. If *destination-list* initially has elements, this procedure adds one of each element in *source-list* to the end of *destination-list*. The content of *source-list* is preserved.

grtl-quicksort-list

Synopsis

grtl-quicksort-list
(*unsorted-list*: text-list, *start-index*: integer, *end-index*: integer)

Argument	Description
unsorted-list	The list to sort.
start-index	The starting index from which to sort, where 0 is the first element.
end-index	The end index for the sort. To sort to the end of the list, specify an index of one less than the length of the list.

Description

Uses a quicksort algorithm to sort a text list alphabetically, either for the entire list or for a subsection of the list.

grtl-remove-subset-list

Synopsis

grtl-remove-subset-list
    (*superset*: item-list, *subset*: item-list)

Argument	Description
superset	The item-list from which to remove items.
subset	The list of items to remove from *superset*.

Description

Removes the items contained in *subset* from the items in *superset*.

6

Vectors

Describes the vector class and methods for manipulating vectors.

Introduction

GRTL provides a vector class, which is a subclass of g2-list that manages the size of the list by removing elements, as needed. The vector class defines methods for adding, removing, and deleting elements, getting elements, and getting and setting the vector size.

Classes

grtl-vector

grtl-vector

A g2-list for managing values or references. The vector size can be bound to a maximum size. If new items are added and the number of elements exceeds the maximum size of the vector, the first element is removed and the new element is added at the end (FIFO style).

Class Inheritance Path

grtl-vector, g2-list, object, item

Methods

grtl-vector::grtl-add-element

Synopsis

grtl-vector::grtl-add-element
(*vector*: class grtl-vector, *element*: item-or-value)

Argument	Description
vector	The vector.
element	The element to add.

Description

Adds an element to the end of a vector. The size of the vector cannot be exceeded. If the maximum size is reached when a new element is added, the first element is removed (FIFO style) if it's a scalar; otherwise, the first element is deleted if the element is an item. Therefore, the size of the vector is maintained constant.

grtl-vector::grtl-delete-all-elements

Synopsis

grtl-vector::grtl-delete-all-elements
    (*vector*: class grtl-vector)

Argument	Description
vector	The vector.

Description

Deletes all elements from a vector if the elements are items; otherwise, removes the elements from the vector if they are scalar elements.

grtl-vector::grtl-delete-first-element

Synopsis

grtl-vector::grtl-delete-first-element
    (*vector*: class grtl-vector)

Argument	Description
vector	The vector.

Description

Deletes the first element in a vector, which is the oldest element. If the first element is an item, it also deletes the item. If the vector is empty, this method does nothing.

grtl-vector::grtl-delete-last-element

Synopsis

grtl-vector::grtl-delete-last-element
    (*vector*: class grtl-vector)

Argument	Description
vector	The vector.

Description

Deletes the first element in a vector, which is the newest element. If the first element is an item, it also deletes the item. If the vector is empty, this method does nothing.

grtl-vector::grtl-get-element-at

Synopsis

grtl-vector::grtl-get-element-at
    (*vector*: class grtl-vector, *index*: integer)
    -> *element*: item-or-value

Argument	Description
vector	The vector.
index	The position.

Return Value	Description
element	The resulting element.

Description

Returns the element at a specified index of a vector. If the index is not valid, this method signals a grtl-invalid-index error.

grtl-vector::grtl-get-head-element

Synopsis

grtl-vector::grtl-get-head-element
    (*vector*: class grtl-vector)
    -> *element*: item-or-value

Argument	Description
vector	The vector.

Return Value	Description
element	The first element.

Description

Returns the first element in a vector, which is the oldest element, provided the vector has reached its maximum size; otherwise, returns the symbol null.

grtl-vector::grtl-get-size

Synopsis

grtl-vector::grtl-get-size
    (*vector*: class grtl-vector)
    -> *size*: integer

Argument	Description
vector	The vector.

Return Value	Description
size	The size of the vector.

Description

Returns the maximum size of a vector.

grtl-vector::grtl-number-of-elements

Synopsis

grtl-vector::grtl-number-of-elements
    (*vector*: class grtl-vector)
    -> *count*: integer

Argument	Description
vector	The vector.

Return Value	Description
count	The number of elements.

Description

Returns the number of elements in a vector.

grtl-vector::grtl-remove-all-elements

Synopsis

grtl-vector::grtl-remove-all-elements
(*vector*: class grtl-vector)

Argument	Description
vector	The vector.

Description

Removes all elements from vector. If the elements are items, the items are not deleted.

grtl-vector::grtl-remove-element

Synopsis

grtl-vector::grtl-remove-element
    (*vector*: class grtl-vector, *element*: item-or-value)

Argument	Description
vector	The vector.
element	The element to remove.

Description

Removes an element from a vector, if it actually is an element of the vector. If the specified element is not in the vector, this method does nothing.

grtl-vector::grtl-remove-first-element

Synopsis

grtl-vector::grtl-remove-first-element
(*vector*: class grtl-vector)

Argument	Description
vector	The vector.

Description

Removes the first element in the vector, which is the oldest element. If the elements are items, the items are not deleted. If the vector is empty, this method does nothing.

grtl-vector::grtl-remove-last-element

Synopsis

grtl-vector::grtl-remove-last-element
    (*vector*: class grtl-vector)

Argument	Description
vector	The vector.

Description

Removes the last element in the vector, which is the newest element. If the elements are items, the items are not deleted. If the vector is empty, this method does nothing.

grtl-vector::grtl-set-size

Synopsis

grtl-vector::grtl-set-size
(*vector*: class grtl-vector, *size*: integer)

Argument	Description
vector	The vector.
size	The maximum number of elements.

Description

Sets the maximum size of the vector.

7

Sequences

Describes procedures to manipulate sequences.

Introduction

GRTL provides procedures for manipulating sequences, including:

- Determining if two sequences are identical.
- Determining if elements in one sequence are included in another sequence.
- Generating an empty sequence.

- Inverting the elements of a sequence.
- Removing duplicate values from a sequence.
- Removing objects and values from a sequence.
- Replacing values in a sequence.
- Sorting a sequence.

grtl-are-sequences-identical

Synopsis

grtl-are-sequences-identical
    (seq1: sequence, seq2: sequence)
    -> *result*: truth-value

Argument	Description
seq1	The first sequence.
seq2	The second sequence.

Return Value	Description
result	True if the sequences are identical, false otherwise.

Description

Returns true if two sequences are identical, false otherwise.

Two sequences are identical if they contain the same elements and the same number of elements. The order of the elements may be different, e.g., `sequence(a, b, c)` is treated as identical to `sequence(c, b, a)`.

To compare sequences that are strictly identical (same order of elements), use "`=`" instead.

grtl-check-sequence-inclusion

Synopsis

grtl-check-sequence-inclusion
    (*seq1*: sequence, *seq2*: sequence)
    -> *included*: truth-value

Argument	Description
seq1	The first sequence.
seq2	The second sequence.

Return Value	Description
included	True if all elements of the first sequence are included in the second sequence, false otherwise.

Description

Returns true if each element of *seq1* is also an element of *seq2*; otherwise, returns false.

grtl-invert-sequence

Synopsis

grtl-invert-sequence
    (*source-sequence*: sequence)
    -> *seq*: sequence

Argument	Description
source-sequence	The sequence to invert.

Return Value	Description
seq	The inverted sequence.

Description

Inverts the contents of the source sequence.

grtl-remove-duplicate-values-from-sequence

Synopsis

grtl-remove-duplicate-values-from-sequence
    (*source-sequence*: sequence)
    -> *seq*: sequence

Argument	Description
source-sequence	The source sequence.

Return Value	Description
seq	The resulting sequence.

Description

Removes duplicate values (not items) from a sequence and returns the resulting sequence.

grtl-remove-first-value-from-sequence

Synopsis

grtl-remove-first-value-from-sequence
(*source-sequence*: sequence, *val*: value)
-> *seq*: sequence

Argument	Description
source-sequence	The source sequence.
val	The value to remove.

Return Value	Description
seq	The resulting sequence.

Description

Removes the first occurrence of a value in a sequence and returns the resulting sequence.

grtl-remove-object-from-sequence

Synopsis

grtl-remove-object-from-sequence
    (*source-sequence*: sequence, *obj*: object)
    -> *seq*: sequence

Argument	Description
source-sequence	The source sequence.
obj	The object to remove.

Return Value	Description
seq	The resulting sequence.

Description

Removes the first occurrence of an object in a sequence and returns the resulting sequence.

grtl-remove-value-from-sequence

Synopsis

grtl-remove-value-from-sequence
    (*source-sequence*: sequence, *val*: value)
    -> *seq*: sequence

Argument	Description
source-sequence	The source sequence.
val	The value to remove.

Return Value	Description
seq	The resulting sequence.

Description

Removes all occurrences of a value in a sequence and returns the resulting sequence.

grtl-replace-first-value-in-sequence

Synopsis

grtl-replace-first-value-in-sequence
(*source-sequence*: sequence, *old-val*: value, *new-val*: value)
-> *seq*: sequence

Argument	Description
source-sequence	The source sequence.
old-val	The value to remove.
new-val	The new value.

Return Value	Description
seq	The resulting sequence.

Description

Replaces the first occurrence of *old-val* with *new-val* in a sequence and returns the resulting sequence.

grtl-replace-value-in-sequence

Synopsis

grtl-replace-value-in-sequence
(*source-sequence*: sequence, *old-val*: value, *new-val*: value)
-> *seq*: sequence

Argument	Description
source-sequence	The source sequence.
old-val	The value to remove.
new-val	The new value.

Return Value	Description
seq	The resulting sequence.

Description

Replaces all occurrences of *old-val* with *new-val* in a sequence and returns the resulting sequence.

grtl-sort-sequence

Synopsis

grtl-sort-sequence
(*key-sequence*: sequence, *aux-sequence*: item-or-value,
key-function: item-or-value, *comparison-function*: item-or-value,
add-args: structure)
-> *seq1*: sequence, *seq2*: item-or-value

Argument	Description
key-sequence	Specifies the sequence to sort. This is the only required argument.
aux-sequence	An optional, secondary sequence to sort. If you are not using an auxiliary sequence, use false for this argument.
key-function	A user-defined function to apply to the *key-sequence*. If you do not provide a function, use false for this argument.
comparison-function	A user-defined function to apply as a comparison function to the *key-sequence*. If you do not provide a comparison function, use false for this argument.
add-args	A two-attribute structure indicating the sorting order and whether to allow other processing during the procedure. The syntax of the structure is: structure (direction: symbol, allow-other-processing: truth-value) The options for direction are: ascending and descending.

Return Value	Description
seq1	The sorted *key-sequence*.
seq2	The sorted *aux-sequence*, if specified; otherwise, false.

Description

Sorts elements in one or two sequences. This procedure is similar to the g2-sort-list system procedure. For the details on using this procedure, see the description of g2-sort-list in Chapter 24 “Sorting Operations” in the *G2 System Procedures Reference Manual*.

8

Timers

Describes a timer class and procedures to create timers and get the clock time.

Introduction

GRTL provides a timer capability whose clock time continually increases over time, as well as procedures for creating new timers, getting existing timers with a given update interval, and getting the current clock time of a timer.

Classes

grtl-timer

grtl-timer

A timer capability, for example, a 10-second grtl-timer that receives a value every 10 seconds. The value of a timer continues to increase over time; it does not go from *t* to zero over and over again. Waiting until a timer receives a value within a repeat loop guarantees a certain time between iterations, whereas using a wait statement for *n* seconds does not due to the extra time it takes to perform the other operations in the repeat loop.

To get a timer, call grtl-get-timer or grtl-get-new-timer. To use a timer, use wait until T receives a value where T is the timer.

Note that a grtl-timer runs with a procedure priority of 2.

Class Inheritance Path

grtl-timer, float-parameter, quantitative-parameter, parameter, variable-or-parameter, object, grtl-item-with-key, grtl-item, item

Attributes

Attribute	Description
grtl-timer-setting	Set to the timer interval you request in your call to grtl-get-timer or grtl-get-new-timer. The interval is the number of seconds between updates of the timer.
Allowable values:	Any integer
Default value:	0
grtl-poll-interval	Set to the grtl-timer-setting/10 when you create a timer. The interval is the number of seconds between internal polls, which determines if the timer is due for another update. This is also the error per timer update. This error does not accumulate over updates.
Allowable values:	Any float
Default value:	0.0

grtl-start-time

Set to the G2 time at which the timer was started. By comparing this time against the current timer value, which you get by calling grtl-clock-time, and the grtl-timer-setting attribute, you can determine the time remaining until the next update, and how long the timer has been running.

Allowable values: Any integer

Default value: 0

key

See grtl-item-with-key.

Allowable values: inherited

Default value: ""

Procedures

grtl-get-new-timer

Synopsis

grtl-get-new-timer
    (*timer-setting*: integer)
    -> *timer*: class grtl-timer

Argument	Description
timer-setting	The interval in seconds between updates of the timer.

Return Value	Description
timer	A new timer with the requested setting.

Description

Creates a new timer with the requested timer setting, then starts the timer and returns the new timer. This procedure signals a grtl-invalid-interval error if the *timer-setting* is less than 1.

grtl-get-timer

Synopsis

grtl-get-timer
    (*timer-setting*: integer)
    -> *timer*: class grtl-timer

Argument	Description
timer-setting	The interval in seconds between updates of the timer.

Return Value	Description
timer	A new or existing timer with the requested setting.

Description

Returns an existing timer with the requested setting that is already running, if one exists. Otherwise, this procedure creates a new timer, starts the timer, and returns the new timer. To create a new timer unconditionally, use grtl-get-new-timer.

Functions

grtl-clock-time

grtl-clock-time

Synopsis

grtl-clock-time
    (*timer*: class grtl-timer)

Argument	Description
timer	A grtl-timer whose clock time to return.

Description

Returns the current clock time of a timer, rounded to the nearest second. Note that a grtl-timer continuously increases over time.

9

Time

Describes procedures and functions for converting numeric and textual representations of time and durations.

Introduction

GRTL provides procedures for converting time and G2 durations from numeric representations to textual, XML, ISO, and ISOX representations, as well as for converting textual representations of time and durations to numeric values.

Procedures

grtl-day-to-integer

Synopsis

grtl-day-to-integer
    (*day-symbol*: symbol)
    -> *day-int*: integer

Argument	Description
day-symbol	A symbol for the day of the week.

Return Value	Description
day-int	The integer equivalent of the day of the week.

Description

Converts the full English word for a day of the week, in symbol form, to an integer, where 0 is Sunday, 1 is Monday, 2 is Tuesday, and so on, through 6, which is Saturday.

grtl-duration-to-g2-interval

Synopsis

grtl-duration-to-g2-interval
    (*duration-text*: text)
    -> *g2-interval*: integer

Argument	Description
duration-text	A time interval expressed in weeks and smaller units, as a text.

Return Value	Description
g2-interval	The G2 time interval equivalent, in seconds, of the specified duration.

Description

Returns a G2 interval as an integer, in seconds, from a duration text formatted as P*n*W*n*D*n*H*n*M*n*S, where unused values are absent. For example, an input of "P1M1S" (1 minute and 1 second) returns 61 (seconds), and "P1W" (1 week) returns 604800 (seconds).

Because months are not expressed, the uppercase T familiar to ISO users is not used.

grtl-english-ordination

Synopsis

grtl-english-ordination
    (*number*: integer)
    -> *ordinal*: text

Argument	Description
number	The number.

Return Value	Description
ordinal	The ordinal for the specified number.

Description

Returns an ordinal, as text, given an integer. For example, an input of 1 returns "1st", an input of 2 returns "2nd", and so forth. This procedure also handles the English peculiarities of 11, 12, and 13.

grtl-g2-day-integer-to-iso-day

Synopsis

grtl-g2-day-integer-to-iso-day
    (*day-integer*: integer)
    -> *iso-day*: text

Argument	Description
day-integer	A day of the week, as an integer.

Return Value	Description
iso-day	The ISO equivalent of the specified day.

Description

Returns a text string with the day of the week formatted as ---DD. For example, an input of 27 returns "---27".

grtl-g2-interval-to-duration

Synopsis

grtl-g2-interval-to-duration
    (*g2-interval*: quantity)
    -> *duration-text*: text

Argument	Description
g2-interval	A G2 time interval, in seconds.

Return Value	Description
duration-text	The specified time interval expressed in weeks and smaller units, as a text.

Description

Returns a duration text formatted as P*n*W*n*D*n*H*n*M*n*S from an integer, in seconds, where unused values are absent. If only time increments are present, 0D is inserted for ISO consistency. For example, an input of 61 returns "P0D1M1S", and an input of 3600 returns "P0D1H".

Because months are not expressed, the uppercase T familiar to ISO users is not used.

grtl-g2-interval-to-xml-days

Synopsis

grtl-g2-interval-to-xml-days
(*g2-interval*: quantity)
-> *xml-day*: text

Argument	Description
g2-interval	A G2 time interval, in seconds.

Return Value	Description
xml-day	The specified time interval expressed in days, as a text.

Description

Returns a text string with time formatted as P*n*D, where partial days are ignored. For example, an input of 604801 (1 week and 1 second) returns "P7D", thereby ignoring the seconds.

grtl-g2-interval-to-xml-weeks

Synopsis

grtl-g2-interval-to-xml-weeks
(*g2-interval*: quantity)
-> *xml-week*: text

Argument	Description
g2-interval	A G2 time interval, in seconds.

Return Value	Description
xml-week	The specified time interval expressed in weeks, as a text.

Description

Returns a text string with time formatted as P*n*W, where partial weeks are ignored. For example, an input of 604801 (1 week and 1 second) returns "P1W", thereby ignoring the seconds.

grtl-g2-month-integer-to-iso-month

Synopsis

grtl-g2-month-integer-to-iso-month
(*month-int*: integer)
-> *iso-month*: text

Description

Argument	Description
day-integer	A month expressed as an integer.

Return Value	Description
iso-month	The ISO equivalent of the specified month.

Description

Returns a text string with the month formatted as ---MM. For example, an input of 7 returns "--07".

grtl-g2-time-from-text-mm-dd-yy-hh-mm-ss

Synopsis

grtl-g2-time-from-text-mm-dd-yy-hh-mm-ss
(*duration-text*: text)
-> *g2-time*: integer

Argument	Description
duration-text	A date and time, as a text.

Return Value	Description
g2-time	The G2 time equivalent of the specified duration, in seconds.

Description

Returns the integer timestamp given a duration text formatted as "mm dd yy hh mm ss". Any delimiter is accepted, and all fields can have any number of consecutive digits. In the case of two-digit years, the twentieth century is assumed for years 70 and above.

For example, assuming G2 was started on January 1, 2004 12:00:00, given an input of "1 1 04 12 0 0" (January 2, 2004 12:00:00), this procedure returns 86400 (24 hours).

grtl-g2-time-from-text-yyyy-dd-mm-hh-mm-ss

Synopsis

grtl-g2-time-from-text-yyyy-dd-mm-hh-mm-ss
(*time-text*: text)
-> *g2-time*: integer

Argument	Description
duration-text	A date and time, as a text.

Return Value	Description
g2-time	The G2 time equivalent of the specified duration, in seconds.

Description

Returns the integer timestamp given a duration text formatted as "yyyy dd mm hh mm ss". Any delimiter is accepted, and all fields must have a fixed number of digits, as formatted above.

For example, assuming G2 was started on January 1, 2004 12:00:00, given an input of "2004 01 02 12 00 00" (January 2, 2004 12:00:00), this procedure returns 86400 (24 hours).

grtl-g2-time-to-iso-day

Synopsis

grtl-g2-time-to-iso-day
    (*g2-time*: quantity)
    -> *iso-day*: text

Argument	Description
g2-time	A G2 time interval, in seconds.

Return Value	Description
iso-day	The ISO day equivalent of the specified G2 time.

Description

Returns a text string with the time interval formatted as ---DD. For example, assuming G2 was started on January 1 12:00:00, an input of 86400 (24 hours) returns "---02" (January 2).

grtl-g2-time-to-iso-month

Synopsis

grtl-g2-time-to-iso-month
    (*g2-time*: quantity)
    -> *iso-month*: text

Argument	Description
g2-time	A G2 time interval, in seconds.

Return Value	Description
iso-month	The ISO month equivalent of the specified G2 time.

Description

Returns a text string with the time interval formatted as ---MM. For example, assuming G2 was started on January 31 12:00:00, an input of 86400 (24 hours) returns "---02" (February).

grtl-g2-time-to-iso-year

Synopsis

grtl-g2-time-to-iso-year
    (*g2-time*: quantity)
    -> *iso-year*: text

Argument	Description
g2-time	A G2 time interval, in seconds.

Return Value	Description
iso-year	The ISO year equivalent of the specified G2 time.

Description

Returns a text string with the time interval formatted as YYYY. For example, assuming G2 was started on January 1, 2004 12:00:00, an input of 86400 (24 hours) returns "2004".

grtl-g2-time-to-iso-year-month

Synopsis

grtl-g2-time-to-iso-year-month
(*g2-time*: quantity)
-> *iso-year-month*: text

Argument	Description
g2-time	A G2 time interval, in seconds.

Return Value	Description
iso-year-month	The ISO year and month equivalent of the specified G2 time.

Description

Returns a text string with the time interval formatted as YYYY-MM. For example, assuming G2 was started on January 31, 2004 12:00:00, an input of 86400 (24 hours) returns "2004-02" (February 2004).

grtl-g2-time-to-isox-date

Synopsis

grtl-g2-time-to-isox-date
    (*g2-time*: quantity)
    -> *isox-date*: text

Argument	Description
g2-time	A G2 time interval, in seconds.

Return Value	Description
isox-date	The extended ISO date equivalent of the specified G2 time.

Description

Returns a text string with the time interval formatted as YYYY-MM-DD, the extended ISO date format. For example, assuming G2 was started on January 1, 2004 12:00:00, an input of 86400 (24 hours) returns "2004-01-02" (January 2, 2004).

grtl-g2-time-to-isox-date-time

Synopsis

grtl-g2-time-to-isox-date-time
    (*g2-time*: quantity)
    -> *isox-date-time*: text

Argument	Description
g2-time	A G2 time interval, in seconds.

Return Value	Description
isox-date-time	The extended ISO date-time equivalent of the specified G2 time.

Description

Returns a text string with the time interval formatted as YYYY-MM-DDThh:mm:ss, the extended ISO date-time format. For example, assuming G2 was started on January 1, 2004 12:00:00, an input of 86401 (24 hours and 1 second) returns "2004-01-02T12:00:01" (January 2, 2004 12:00:01).

grtl-g2-time-to-isox-month-day

Synopsis

grtl-g2-time-to-isox-month-day
    (*g2-time*: quantity)
    -> *isox-month-day*: text

Argument	Description
g2-time	A G2 time interval, in seconds.

Return Value	Description
isox-month-day	The extended ISO month-day equivalent of the specified G2 time.

Description

Returns a text string with the time interval formatted as --MM-DD, the extended ISO month-day format. For example, assuming G2 was started on January 1, 2004 12:00:00, an input of 86400 (24 hours) returns "--01-02" (January 2, 2004).

grtl-g2-time-to-isox-time

Synopsis

grtl-g2-time-to-isox-time
(*g2-time*: quantity)
-> *isox-time*: text

Description

Returns a string with time formatted as hh:mm:ss; extended IOS time

Argument	Description
g2-time	A G2 time interval, in seconds.

Return Value	Description
isox-time	The extended ISO time equivalent of the specified G2 time.

Description

Returns a text string with the time interval formatted as hh:mm:ss, the extended ISO time format. For example, assuming G2 was started on January 1, 2004 12:00:00, an input of 60 (1 minute) returns "12:01:00".

grtl-g2-time-to-text

Synopsis

grtl-g2-time-to-text
(*g2-time*: quantity, *format-string*: text)
-> *duration-text*: text

Argument	Description
g2-time	A G2 time interval, in seconds.
format-string	A text string consisting of date and time fields, plus any other text you want to insert.

Return Value	Description
duration-text	The date and time equivalent of the specified G2 time formatted using the specified format string.

Description

Returns a formatted time given a G2 time and format string. The format string can include any of the following:

<y2> 2-digit year

<y4> 4-digit year

<m> month with 1 or 2 digits

<m2> month with 2 digits, zero-filled

<mf> full month name

<ma> abbreviated (3 characters) month name

<d> day with 1 or 2 digits

<d2> day with 2 digits, zero-filled

<dor> ordinal day

<df> full day-of-the-week name

<da> abbreviated (3 characters) day-of-the-week name

<apm> A.M. or P.M.

<h12> hour in 12-hour format

<h24> hour in 24-hour format

<min> minutes with 2 digits, zero-filled

<sec> seconds with 2 digits, zero-filled

For example, a format string of "Date: <m>/<d>, <y4> Time: <h12>:<min>:<sec> <apm>" would result in a duration text that looks like this:
Date: 4/2, 2004 Time: 1:01:04 P.M.

grtl-g2-year-integer-to-iso-year

Synopsis

grtl-g2-year-integer-to-iso-year
    (*year-int*: integer)
    -> *iso-year*: text

Argument	Description
year-integer	A year specified as an integer.

Return Value	Description
iso-year	The ISO equivalent of the specified year.

Description

Returns a text string with the year formatted as yyyy. For example, an input of 2004 returns "2004".

grtl-interval-in-hours-from-text-hh-mm-ss

Synopsis

grtl-interval-in-hours-from-text-hh-mm-ss
(*duration-text*: text)
-> *g2-interval*: float

Argument	Description
duration-text	A time interval expressed as text.

Return Value	Description
g2-interval	The specified time interval expressed in decimal hours, as a float.

Description

Returns a time interval expressed in decimal hours (hh:hh) given a time interval expressed as hh:mm:ss. For example, an input of 12:30:00 returns 12.5.

grtl-interval-in-seconds-from-text-hh-mm

Synopsis

grtl-interval-in-seconds-from-text-hh-mm
(*duration-text*: text)
-> *g2-interval*: integer

Argument	Description
duration-text	A time interval expressed as text.

Return Value	Description
g2-interval	The specified time interval expressed in seconds.

Description

Returns a time interval expressed in seconds given a time interval expressed as HH-MM or hh:mm. For example, an input of 12:00 returns 43200.

grtl-interval-in-seconds-from-text-hh-mm-ss

Synopsis

grtl-interval-in-seconds-from-text-hh-mm-ss
(*hh-mm*: text)
-> *g2-interval*: integer

Argument	Description
duration-text	A time interval expressed as text.

Return Value	Description
g2-interval	The specified time interval expressed in seconds.

Description

Returns a time interval expressed in seconds given a time interval expressed as HH-MM-SS or hh:mm:ss. For example, an input of 12:00:00 returns 43200.

grtl-isox-date-time-to-g2-time

Synopsis

grtl-isox-date-time-to-g2-time
(*date-time-isox*: text)
-> *g2-time*: float

Argument	Description
date-time-isox	A time interval expressed as the extended ISO date-time.

Return Value	Description
g2-time	The G2 time equivalent of the specified date-time, as a float.

Description

Returns the G2 time from an extended ISO date-time formatted as YYYY-MM-DDThh:mm:ss. For example, assuming G2 was started January 1, 2004 12:00:00, an input of "2004-01-02T12:00:00" (January 2, 12:00:00) returns 86400.

grtl-isox-date-to-g2-time

Synopsis

grtl-isox-date-to-g2-time
(*date-isox*: text)
-> *g2-time*: float

Argument	Description
date-isox	A time interval expressed as the extended ISO date.

Return Value	Description
g2-time	The G2 time equivalent of the specified date, as a float.

Description

Returns the G2 time from an extended ISO date formatted as YYYY-MM-DD. For example, assuming G2 was started January 1, 2004 12:00:00, an input of "2004-01-02" (January 2, 12:00:00) returns 86400.

grtl-minutes-to-hh-mm

Synopsis

grtl-minutes-to-hh-mm
    (*mins*: integer, *mod24*: truth-value)
    -> *duration-text*: text

Argument	Description
mins	A specified number of minutes, as an integer.
mod24	True to return multiples of 24 hours, or false to return the actual number of hours.

Return Value	Description
duration-text	The time interval equivalent of the specified number of minutes, as a text.

Description

Returns a text formatted as hh:mm given a specified number of seconds. For example, an input of 1500 (25 hours) returns 01:00 when *mod24* is true, and 25:00 when *mod24* is false.

grtl-text-date-and-time-from-g2-time

Synopsis

grtl-text-date-and-time-from-g2-time
(*g2-time*: quantity, *format-code*: symbol, *d-delim*: text, *t-delim*: text, *dt-delim*: text)
-> *date-time-text*: text

Argument	Description
g2-time	A G2 time.
format-code	A data-and-time format code: year-month-day-hour-mm-ss month-day-year-hour-mm-ss day-month-year-hour-mm-ss isox-date-time hour-mm-ss hour-mm:hh.hh hh.hh dd.dd g2-time-stamp
d-delim	The delimiter to use between the date fields, for example: '-'
t-delim	The delimiter to use between the time fields, for example: ':'
d-tdelim	The delimiter to use between the date and time fields, for example: ' '

Return Value	Description
date-time-text	The date and time formatted as specified.

Description

Converts a G2 time, which may be a subsecond time, into a date and time text. You can specify one of three formats, and you can specify the delimiters to use.

grtl-text-date-and-time-to-g2-time

Synopsis

grtl-text-date-and-time-to-g2-time
(*date-time-text*: text, *format-code*: symbol)
-> *g2-time*: quantity

Argument	Description
date-time-text	The date and time formatted as a text.
format-code	A data-and-time format code: year-month-day-hour-mm-ss month-day-year-hour-mm-ss day-month-year-hour-mm-ss isox-date-time hour-mm-ss hour-mm:hh.hh hh.hh dd.dd g2-time-stamp

Return Value	Description
g2-time	The G2 time.

Description

Converts a date and time text into G2 time. You can specify one of several formats to use.

grtl-text-date-from-g2-time

Synopsis

grtl-text-date-from-g2-time
(*g2-time*: quantity, *format-code*: symbol, *d-delim*: text)
-> *date-text*: text

Argument	Description
g2-time	A G2 time.
format-code	A date format code: yyyy-mm-dd, mm-dd-yyyy, or dd-mm-yyyy.
d-delim	The delimiter to use between the date fields, for example: '-'

Return Value	Description
date-text	The date formatted as specified.

Description

Converts a G2 time, which may be a subsecond time, into a date text. You can specify one of three formats, and you can specify the delimiters to use.

grtl-text-time-from-g2-time

Synopsis

grtl-text-time-from-g2-time
(*g2-time*: quantity, *format-code*: symbol, *t-delim*: text)
-> *time-text*: text

Argument	Description
g2-time	A G2 time.
format-code	A time format code: hh-mm-ss or hh-mm.
t-delim	The delimiter to use between the time fields, for example: ':'

Return Value	Description
date-text	The time formatted as specified.

Description

Converts a G2 time, which may be a subsecond time, into a time text. You can specify one of three formats, and you can specify the delimiters to use.

grtl-time-in-hours-from-text-mm-dd-yyyy-hh-mm-ss

Synopsis

grtl-time-in-hours-from-text-mm-dd-yyyy-hh-mm-ss
(*time-text*: text)
-> *hours*: float

Argument	Description
time-text	A time interval formatted as text.

Return Value	Description
hours	The G2 time expressed in decimal hours, as a float.

Description

Returns the G2 time in hours, as a quantity, given a time text formatted as MM/DD/YYYY hh:mm:ss. For example, assuming G2 was started January 1, 2004 12:00:00, an input of "01/02/2004 12:00:00" (January 2, 12:00:00) returns 86400.

grtl-xml-days-to-g2-interval

Synopsis

grtl-xml-days-to-g2-interval
    (*days-xml*: text)
    -> *g2-interval*: integer

Argument	Description
days-xml	A time interval expressed in XML days.

Return Value	Description
g2-interval	The specified time interval expressed in seconds.

Description

Returns a G2 time interval, in seconds, from a time interval formatted in XML days as P*n*D. For example, an input of P7D (7 days) returns 604800 (seconds).

grtl-xml-weeks-to-g2-interval

Synopsis

grtl-xml-weeks-to-g2-interval
    (*weeks-xml*: text)
    -> *g2-interval*: integer

Argument	Description
weeks-xml	A time interval expressed in XML weeks.

Return Value	Description
g2-interval	The specified time interval expressed in seconds.

Description

Returns a G2 time interval, in seconds, from a time interval formatted in XML weeks as P*n*W. For example, an input of P1W (1 week) returns 604800 (seconds).

Functions

grtl-mmddyyyyhhmmss-text-to-time

Synopsis

grtl-mmddyyyyhhmmss-text-to-time
(*date-time-text*: text)
-> *g2-time*: quantity

Argument	Description
date-time-text	A date and time interval expressed as a text.

Return Value	Description
g2-time	The G2 time equivalent of the specified date and time, as a quantity.

Description

Returns the G2 time given a date and time formatted as mm/dd/yyyy hh:mm:ss. For example, assuming G2 was started January 1, 2004 12:00:00, an input of "01-02-2004 12:00:00" (January 2, 12:00:00) returns 86400.

grtl-text-time-to-g2-time

Synopsis

grtl-text-time-to-g2-time
(*time-text*: text)
-> *g2-time*: quantity

Argument	Description
time-text	A time interval formatted as text.

Return Value	Description
g2-time	The G2 time equivalent of the specified time text, as a quantity.

Description

Returns the G2 time, as a quantity, given a time text formatted as YYYYMMDDhhmmss. Note that there are no delimiters. For example, assuming G2 was started January 1, 2004 12:00:00, an input of "20040102120000" (January 2, 12:00:00) returns 86400.

grtl-timestamp-to-mmddhhm

Synopsis

grtl-timestamp-to-mmddhhm
(*timestamp*: text)
-> *time-text*: text

Argument	Description
timestamp	A timestamp formatted as text.

Return Value	Description
time-text	The timestamp formatted as a time interval.

Description

Returns a time text formatted as YYYY/MM/DD hh:m1 given a timestamp formatted as YYYYMMDDhhmmss, where m1 means whole minutes. For example, an input of 20040101125900 (January 1, 2004 12:59:00) returns 2004/01/01 12:00.

10

Math

Describes procedures and functions for mathematical operations.

Introduction

GRTL provides a class and methods, procedures, and functions for:

- Defining an object for storing mathematical distributions and their arguments.
- Methods for getting random values from a distribution mode.
- Calculating random values from these mathematical distributions: normal, exponential, triangular, uniform, beta, gamma, lognormal, Weibul, and Erlang.
- Calculating the rate of change and standard deviations of a variable or parameter.
- Returning the values of ε and π.

Classes

grtl-distribution-mode

grtl-distribution-mode

A class with place holders to describe many mathematical distributions and store their arguments. Use grtl-get-float-random-value and grtl-get-integer-random-value to calculate a random sample, based on specification in instances of this class.

Class Inheritance Path

grtl-distribution-mode, object, item

Attributes

Attribute	Description
distribution-type	Specifies the mathematical distribution to use.
Allowable values:	FIXED, EXPONENTIAL, NORMAL, UNIFORM, TRIANGULAR, ERLANG, WEIBEL, LOGNORMAL, GAMMA, BETA
Default value:	NORMAL
mean	
Allowable values:	Any item or value
Default value:	1
standard-deviation	
Allowable values:	Any item or value
Default value:	0
min	
Allowable values:	Any item or value
Default value:	3600
max	
Allowable values:	Any item or value
Default value:	3600
mode	

Allowable values: Any item or value

Default value: 3600

alpha

Allowable values: Any item or value

Default value: 0

beta

Allowable values: Any item or value

Default value: 0

shape

Allowable values: Any item or value

Default value: 0

scale

Allowable values: Any item or value

Default value: 0

number-of-samples

Allowable values: Any integer

Default value: 0

Methods

grtl-distribution-mode::grtl-get-float-random-value
grtl-distribution-mode::grtl-get-integer-random-value

Methods and Procedures

Methods

grtl-distribution-mode::grtl-get-float-random-value
grtl-distribution-mode::grtl-get-integer-random-value
variable-or-parameter::grtl-calculate-rate-of-change
variable-or-parameter::grtl-calculate-standard-deviation

Procedures

grtl-get-random-beta-value
grtl-get-random-erlang-value
grtl-get-random-exponential-value
grtl-get-random-gamma-value
grtl-get-random-lognormal-value
grtl-get-random-normal-value
grtl-get-random-triangular-value
grtl-get-random-uniform-value
grtl-get-random-value-from-text-specification
grtl-get-random-weibel-value

grtl-distribution-mode::grtl-get-float-random-value

Synopsis

grtl-distribution-mode::grtl-get-float-random-value
(*spec*: grtl-distribution-mode)
-> *value*: float

Argument	Description
spec	The distribution mode specification.

Return Value	Description
value	The sample value.

Description

Returns a random float value, based on the distribution mode specification.

grtl-distribution-mode::grtl-get-integer-random-value

Synopsis

grtl-distribution-mode::grtl-get-integer-random-value
    (*spec*: grtl-distribution-mode)
    -> *value*: integer

Argument	Description
spec	The distribution mode specification.

Return Value	Description
value	The sample value.

Description

Returns a random integer value, based on the distribution mode specification.

variable-or-parameter::grtl-calculate-rate-of-change

Synopsis

variable-or-parameter::grtl-calculate-rate-of-change
(*var-or-param*: variable-or-parameter, *roc-units*: integer, *delta-time*: integer, *start-time*: integer, *client*: object)
-> *value*: float

Argument	Description
var-or-param	The variable or parameter whose history is to be analyzed.
roc-units	The number of seconds representing the units of the rate of change, for example, 1 for ROC per second or 60 for ROC per minute.
delta-time	The number of seconds backwards in time from the start time over which to perform the ROC calculation.
start-time	The starting point of the calculation in seconds. Use the expression the current time to start from now.
client	The client of this invocation, typically a G2 window.

Return Value	Description
value	The rate of change of the given variable or parameter.

Description

Calculates the rate of change for a variable or parameter with history. The error grtl-insufficient-history is signalled if less then 2 values exist in the variable or parameter's history. Interpolation and extrapolation are used, as needed, in cases where gaps exist in history.

variable-or-parameter::grtl-calculate-standard-deviation

Synopsis

variable-or-parameter::grtl-calculate-standard-deviation
(*var-or-param*: variable-or-parameter, *delta-time*: integer, *start-time*: integer, *client*: object)
-> *value*: float

Argument	Description
var-or-param	The variable or parameter whose history is to be analyzed.
delta-time	The number of seconds backwards in time from the start time over which to perform calculation.
start-time	The starting point of the calculation in seconds. Use expression the current time to start with current value.
client	Client of this invocation, typically a G2 window.

Return Value	Description
value	The standard deviation of the specified variable or parameter.

Description

Calculates the standard deviation for a variable or parameter with history. The error grtl-insufficient-history is signalled if less then 2 values exist in the variable or parameter's history.

grtl-get-random-beta-value

Synopsis

grtl-get-random-beta-value
(*alpha*: float, *beta*: float, *min*: float, *max*: float)
-> *value*: float

Argument	Description
alpha	The alpha of the random beta distribution.
beta	The beta of the random beta distribution.
min	The minimum of the random beta distribution.
max	The maximum of the random beta distribution.

Return Value	Description
value	The sample value.

Description

Returns a random beta sample value.

grtl-get-random-erlang-value

Synopsis

grtl-get-random-erlang-value
    (*mean*: float, *nb-of-samples*: integer)
    -> *value*: float

Argument	Description
mean	The mean of the random erlang distribution.
nb-of-samples	The number of samples to use to calculate the result.

Return Value	Description
value	The sample value.

Description

Returns a random Erlang sample value.

grtl-get-random-exponential-value

Synopsis

grtl-get-random-exponential-value
    (*mean*: float)
    -> *value*: float

Argument	Description
mean	The mean of the random exponential distribution.

Return Value	Description
value	The sample value.

Description

Returns a random exponential sample value.

grtl-get-random-gamma-value

Synopsis

grtl-get-random-gamma-value
    (*alpha*: float, *beta*: float)
    -> *value*: float

Argument	Description
alpha	The alpha of the random gamma distribution.
beta	The beta of the random gamma distribution.

Return Value	Description
value	The sample value.

Description

Returns a random gamma sample value.

grtl-get-random-lognormal-value

Synopsis

grtl-get-random-lognormal-value
(*mean*: float, *standard-deviation*: float, *valid-next-sample*: truth-value, *next-sample*: float)
-> *value*: float, *arg1*: truth-value, *arg2*: float

Argument	Description
mean	The mean of the random lognormal distribution.
standard-deviation	The standard deviation of the random lognormal distribution.
valid-next-sample	True to return the last call to this API as the next sample value or false to return the first call.
next-sample	The last float argument returned by a call to this API or 0.0 for the first call.

Return Value	Description
value	The sample value.
arg1	A truth-value to pass as an argument to the next call to this API.
arg2	A float to pass as an argument to the next call to this API.

Description

Returns a random lognormal sample value.

grtl-get-random-normal-value

Synopsis

grtl-get-random-normal-value
(*mean*: float, *standard-deviation*: float, *valid-next-sample*: truth-value, *next-sample*: float)
-> *value*: float, truth-value, float

Argument	Description
mean	The mean of the random normal distribution.
standard-deviation	The standard deviation of the random normal distribution.
valid-next-sample	True to return the last call to this API as the next sample value or false to return the first call.
next-sample	The last float argument returned by a call to this API or 0.0 for the first call.

Return Value	Description
value	The random normal sample value.
arg1	A truth-value to pass as an argument to the next call to this API.
arg2	A float to pass as an argument to the next call to this API.

Description

Returns a random normal sample value.

grtl-get-random-triangular-value

Synopsis

grtl-get-random-triangular-value
    (*min*: float, *mode*: float, *max*: float)
    -> *value*: float

Argument	Description
min	The minimum value of the random triangular distribution.
mode	The mode value of the random triangular distribution.
max	The maximum value of the random triangular distribution.

Return Value	Description
value	The sample value.

Description

Returns a random triangular sample value.

grtl-get-random-uniform-value

Synopsis

grtl-get-random-uniform-value
    (*min*: float, *max*: float)
    -> *value*: float

Argument	Description
min	The minimum value of the random uniform distribution.
max	The maximum value of the random uniform distribution.

Return Value	Description
value	The sample value.

Description

Returns a random uniform sample value.

grtl-get-random-value-from-text-specification

Synopsis

grtl-get-random-value-from-text-specification
    (*spec*: text)
    -> *value*: float

Argument	Description
spec	The specification of the random value to calculate.

Return Value	Description
value	The sample value.

Description

Parses the specification to extract the random procedure and arguments to use, then uses the other mathematical APIs to calculate the sample value. The generic syntax of the specification is:

    "*keyword*(arg1, arg2, ...)"

The number of arguments depend on the mathematical distribution. All arguments should be floating point numbers, with the exception of *nb-of-sample*, which should be an integer.

For evaluation purpose, the code only uses the first 3 characters of each keyword to identify it, enabling more efficient entry of the specification.

The options for *keyword* and their arguments are:

    fix(*mean*)
    exp(*mean*)
    nor(*mean*, *std*)
    uni(*min*, *max*)
    tri(*min*, *mode*, *max*)
    erl(*mean*, *nb-of-samples*)
    wei(*scale*, *shape*)
    log(*mean*, *std*)
    gam(*alpha*, *beta*)
    bet(*alpha*, *beta*, *min*, *max*)

grtl-get-random-weibel-value

Synopsis

grtl-get-random-weibel-value
    (*scale*: float, *shape*: float)
    -> *value*: float

Argument	Description
scale	The mean value of the random Weibel distribution.
shape	The shape value of the random Weibel distribution.

Return Value	Description
value	The sample value.

Description

Returns a random Weibel sample value.

Functions

e

Synopsis

e
( )

Description

Returns the value of ε (2.71828182845904523536028 7). Note that e is a function, not a procedure.

pi

Synopsis

pi
( )

Description

Returns the value of π (3.14159265358979323846264317 4). Note that pi is a function, not a procedure.

11

Text Manipulation

Describes procedures and functions for manipulating text and converting text to symbols.

Introduction

GRTL provides procedures and functions for manipulating text, including:

- Padding, unpadding, and truncating strings.
- Formatting symbols into text.
- Removing carriage returns, leading and trailing spaces, and quotes.
- Searching and replacing text.
- Converting text to a symbol.
- Converting comma-delimited text to a sequence.
- Comma-delimiting quantities.
- Right-aligning text.

Procedures

grtl-evaluate-query-for-object

Synopsis

grtl-evaluate-query-for-object
(*query*: text, *obj-or-struct*: item-or-value, *input-object-name*: symbol)
-> *query-text*: text

Argument	Description
query	The string with query patterns starting with $ or in brackets.
obj-or-struct	The target object or structure from which to extract values.
input-object-name	The name used in the query to evaluate that is referring to *obj-or-struct* in expressions in brackets.

Return Value	Description
query-text	The updated query.

Description

Takes an item or structure and a query string, and returns a text with attribute names and queries replaced by their value. Queries can be denoted by a $ (for example, $myatt or $myatt.subbject.attr) or within brackets. If the query is within brackets, the keyword TargetObject should be used to refer to the *obj-or-struct* passed as an argument. Note that any brackets in the query string need to be escaped with the character, and the expression in bracket needs to be a valid G2 expression; otherwise, the procedure aborts and signal an error.

Here are some examples of text substitution:

- Text substitution, using attributes of subobjects:
  - Key: $Key
  - P Value = $P
  - Key of P = $P.Key
  - T Value = $T
  - Low T Limit = $T.limits.lo-limit

  - High T Limit = $T.limits.hi-limit
  - DP Value = $T.limits.another-dp
  - DP Key = $T.limits.another-dp.key
  - DP Status = $T.limits.another-dp.status
- Text substitution, using expressions:
  - Key: @[the key of TARGET @]
  - P Value = @[the P of TARGET @]
  - Key of P = @[the Key of the P of TARGET @]
  - T Value = @[the T of TARGET @]
  - Low T Limit = @[the lo-limit of the limits of the T of TARGET @]
  - High T Limit = @[the hi-limit of the limits of the T of TARGET @]
  - DP Value = @[the another-dp of the limits of the T of TARGET @]
  - DP Key = @[the key of the another-dp of the limits of the T of TARGET @]
  - DP Status = @[the status of the another-dp of the limits of the T of TARGET @]
- Text substitution, using history expressions:
  - Average: @[the average value of the pv of TARGET during the last 5 minutes]
  - Maximum: @[the maximum value of the pv of TARGET during the last 1 hour]
  - Minimum: @[the minimum value of the pv of TARGET between 48 hours ago and 24 hours ago]
  - Standard Deviation: @[the standard deviation of the pv of TARGET during the last 1 week]
  - Rate of Change: @[the rate of change per minute of the pv of TARGET during the last 1 hour]
- Text substitution that refers to related items:
  - P Value of related item: @[the P of the substitution-test that is a-related-test-item-of TARGET @]
  - Key of P of related item: @[the key of the P of the substitution-test that is a-related-test-item-of TARGET @]

grtl-format-symbol-to-text

Synopsis

grtl-format-symbol-to-text
    (*sym*: symbol)
    -> *result*: text

Argument	Description
sym	The symbol to convert to a text.

Return Value	Description
result	The resulting formatted text.

Description

Converts a symbol to a text string. The procedure replaces the - and _ characters in a string with a space, and capitalizes the resulting sting.

grtl-pad-or-truncate-string

Synopsis

grtl-pad-or-truncate-string
(*input-text*: text, *pad-character*: text, *maximum-string-length*: integer)
-> *output-text*: text

Argument	Description
input-text	The input string to pad or truncate.
pad-character	The character to use as padding.
maximum-string-length	The maximum length of the resulting text string.

Return Value	Description
output-text	The resulting padded or truncated string.

Description

Either pads or truncates a string so that its length becomes a fixed length. For example:

grtl-pad-or-truncate-string(“HI”, “_”, 5) returns "HI___".

grtl-pad-or-truncate-string(“Hi there”, “_”, 5) returns "Hi th".

grtl-remove-cr-from-text

Synopsis

grtl-remove-cr-from-text
    (*input-text*: text)
    -> *output-text*: text

Argument	Description
input-text	The input string from which to remove carriage returns.

Return Value	Description
output-text	The resulting string without carriage return.

Description

Removes every occurrence of a carriage return in a string and replaces it with a space. It also replaces multiple spaces, shrinking down two or more spaces to one.

grtl-remove-enclosing-quotes-from-text

Synopsis

grtl-remove-enclosing-quotes-from-text
    (*input-text*: text)
    -> *output-text*: text

Argument	Description
input-text	The input string from which to remove quotes.

Return Value	Description
output-text	The resulting string without quotes.

Description

Removes any enclosing quote characters (") from the input string and returns the resulting string.

grtl-replace-text

Synopsis

grtl-replace-text
    (*from-text*: text, *to-text*: text, *input-text*: text)
    -> *output-text*: text

Argument	Description
from-text	The text to replace.
to-text	The text to substitute.
input-text	The input string to parse and replace *from-text* with *to-text*.

Return Value	Description
output-text	The resulting string with replaced text.

Description

Replaces every occurrence of *from-text* with the *to-text* in *input-text* and returns the resulting string.

grtl-search-and-replace-character

Synopsis

grtl-search-and-replace-character
(*input-text*: text, *character-to-search*: text, *replacement-character*: text)
-> *output-text*: text

Argument	Description
input-text	The input string whose characters should be replaced.
character-to-search	The search character.
replacement-character	The replacement character.

Return Value	Description
output-text	The resulting string with replaced characters.

Description

Replaces all occurrences of *character-to-search* with *replacement-character* in *input-string* and returns the resulting string. This procedure is executed recursively.

grtl-strip-leading-spaces-from-text

Synopsis

grtl-strip-leading-spaces-from-text
    (*input-text*: text)
    -> *output-text*: text

Argument	Description
input-text	The input text whose spaces to strip.

Return Value	Description
output-text	The resulting string without leading spaces.

Description

Removes the right and left leading spaces of the input string and returns the resulting string.

grtl-strip-leading-and-trailing-spaces-from-text

Synopsis

grtl-strip-leading-and-trailing-spaces-from-text
    (*input-text*: text)
    -> *output-text*: text

Argument	Description
input-text	The input text whose spaces to strip.

Return Value	Description
output-text	The resulting string without leading and trailing spaces.

Description

Removes the right and left leading and trailing spaces of the input string and returns the resulting string.

grtl-strip-trailing-spaces-from-text

Synopsis

grtl-strip-trailing-spaces-from-text
    (*input-text*: text)
    -> *output-text*: text

Argument	Description
input-text	The input string whose spaces to strip.

Return Value	Description
output-text	The resulting string without trailing spaces.

Description

Strips all trailing spaces from the input string and returns the resulting string.

grtl-tokenize-csv-record

Synopsis

grtl-tokenize-csv-record
    (*record*: text)
    -> *seq*: sequence

Argument	Description
record	A comma-separated record of fields.

Return Value	Description
seq	A sequence of texts to use as tokens.

Description

Converts a single record (line) from a comma-separated value file into a sequence of tokens. Each token is a text. An empty string token is created when nothing lies between two commas. Since it is assumed that there will be one less comma than there are fields in the record, an empty string record returns a sequence of one empty string.

grtl-unpad-string

Synopsis

grtl-unpad-string
    (*input-text*: text, *pad-character*: text)
    -> *output-text*: text

Argument	Description
input-text	The input text to unpad.
pad-character	The pad character to remove.

Return Value	Description
output-text	The resulting unpadded string.

Description

Drops all header and trailer characters matching the argument *pad-character*. For instance, if *pad-character* is "#" and *input-string* is "###hello!#", then the resulting string is "=hello!". This procedure is particularly useful for removing blank characters surrounding another string. In cases where all characters match *pad-character*, then the returned string is null (""). If one or more characters matching *pad-character* happen to be embedded into the result string, then those characters are not removed and will be part of the returned string. For example, if *input-string* is "####HELLO # YOU##", then the returned string is "HELLO # YOU".

grtl-uppercase-symbol

Synopsis

grtl-uppercase-symbol
    (*txt*: text)
    -> *sym*: symbol

Argument	Description
txt	Any text.

Return Value	Description
sym	The symbol equivalent in upper case.

Description

The built-in G2 symbol function does not permit you to create symbols with lower-case letters or other special characters while automatically converting them to upper case. This procedure converts any text to a symbol such that if s = call grtl-uppercase-symbol(t), then "[s]" = uppcase(t). This function does not work for the empty string. Note also that this function removes any carriage returns from the input string, removes leading spaces, and replaces spaces with the hyphen (-) character.

Functions

comma-delimit

Synopsis

comma-delimit
    (*qty*: quantity)

Argument	Description
qty	The quantity to delimit.

Description

This function takes a quantity argument and adds in the comma delimiter for 1000s. The quantity must be less than a billion. For example: $1,000,000.

pad-to-2-digits

Synopsis

pad-to-2-digits
    (*int*: integer)

Argument	Description
int	Any integer.

Description

This function pads integers less than 10 to a 2-digit text, where a zero is placed in the left as the first digit.

pad-to-3-digits

Synopsis

pad-to-3-digits
    (*int*: integer)

Argument	Description
int	Any integer.

Description

This function pads integers less than 100 where zeros are placed on the left to make a 3-digit text.

right-align

Synopsis

right-align
    (*txt*: text, *max-length*: integer)

Argument	Description
txt	Any text.
max-length	The maximum length of the text.

Description

This function right-justifies the input text, given a maximum length. The maximum length is 24.

12

KB Modules

Describes procedures for manipulating KB modules.

Introduction

GRTL provides procedures for managing modules that:

- Add and remove modules and module references.
- Create, delete, and rename modules.
- Generate a list of modules.
- Get and set modules by name.
- Determine whether a module exists.

grtl-add-module-to-hierarchy-top

Synopsis

grtl-add-module-to-hierarchy-top
    (*module-name*: symbol)

Argument	Description
module-name	The name of the module to add.

Description

Adds the given module to the directly-required-modules of the top-level module, if it is not already present and if it is not the same as the top-level module.

grtl-assign-stray-workspaces-to-top

Synopsis

grtl-assign-stray-workspaces-to-top
( )

Description

Assigns any workspace that is not assigned to a G2 module to the top-level module.

grtl-create-module

Synopsis

grtl-create-module
    (*module-name*: symbol, *parent-module-name*: symbol)

Argument	Description
module-name	The name of the module to create.
parent-module-name	The name of the parent module.

Description

Creates a new module and adds it into the list of directly-required-modules of the parent module. If the parent module does not exist, the top-level module is assigned as the parent module.

grtl-delete-module

Synopsis

grtl-delete-module
(*module-name*: symbol, *delete-workspaces*: truth-value,
update-hierarchy: truth-value)

Argument	Description
module-name	The name of the module to delete.
delete-workspaces	If true, deletes all workspaces assigned to the module.
update-hierarchy	If true, updates the module hierarchy.

Description

Deletes a module and optionally deletes all workspaces assigned to the module and updates the module hierarchy.

grtl-get-module-by-name

Synopsis

grtl-get-module-by-name
    (*module-name*: symbol)
    -> *module-info*: item-or-value

Argument	Description
module-name	The name of the module to get.

Return Value	Description
module-info	Returns the module-information of the specified module; otherwise, returns false if the module is not found.

Description

Searches and return the module-information object for the given module or false if it doesn't exist

grtl-module-exists-for-file

Synopsis

grtl-module-exists-for-file
    (*filename*: text)
    -> *module-info*: class module-information

Argument	Description
filename	The filename of a module.

Return Value	Description
module-info	Returns the module-information of the specified module.

Description

Returns the module-information object for the specified filename. If the module does not exist, it signals the grtl-g2-module-does-not-exist-for-file error.

grtl-remove-module-references

Synopsis

grtl-remove-module-references
    (*module-name*: symbol)

Argument	Description
module-name	The name of the module to remove.

Description

Removes the specified module from the directly-required-modules of every module in the KB. This procedure is useful when deleting a module to keep the module hierarchy properly modularized.

grtl-rename-module

Synopsis

grtl-rename-module
    (*current-module-name*: symbol, *new-module-name*: symbol)

Argument	Description
current-module-name	The name of a module to rename.
new-module-name	The name of the new module.

Description

Renames a module and ensures that all workspaces assigned to the module are correctly assigned after the name changed.

grtl-replace-module-references

Synopsis

grtl-replace-module-references
(*module-name*: symbol, *new-module*: symbol)

Argument	Description
module-name	The name of the old module.
new-module	The name of the new module.

Description

Replaces all references to the given module name with a reference to a new module name in the directly-required-modules of all KB modules. This procedure is useful when renaming a module to keep the module hierarchy properly modularized.

grtl-set-module-file-name

Synopsis

grtl-set-module-file-name
    (*module-name*: symbol, *filename*: text)

Argument	Description
module-name	The name of a module whose filename to set.
filename	A filename for the module.

Description

Changes the module-file-name of the given module to a new filename. If the module does not exist, it signals the grtl-g2-module-does-not-exist error.

13

Files and Directories

Describes procedures for manipulating file names and directory names.

Introduction

GRTL provides procedures for manipulating file names and directory names, including:

- Converting strings to valid file names.
- Creating and deleting directories.
- Extracting file name and directory name from path names.
- Formatting file names by removing spaces and converting to upper case.
- Converting file names to path names.
- Getting the default directory, installation directory, and application directory.
- Getting the file system delimiter.
- Localizing path names.

grtl-convert-string-to-valid-filename

Synopsis

grtl-convert-string-to-valid-filename
    (*string*: text)
    -> *filename*: text

Argument	Description
string	The text string to convert.

Return Value	Description
filename	The resulting filename.

Description

Converts a text string to a valid filename, replacing any invalid characters in the string with the "_" character.

grtl-create-directory

Synopsis

grtl-create-directory
    (*pathname*: text)

Argument	Description
pathname	The pathname to create.

Description

Creates the specified directory, including any parent directories, if required, by calling g2-create-directory. After creating the directory, the procedure tests if the directory exists by calling g2-directory-exists and signals the error grtl-could-not-create-directory if it does not exist.

grtl-delete-directory

Synopsis

grtl-delete-directory
    (*pathname*: text)

Argument	Description
pathname	The pathname to delete.

Description

Deletes a directory. After deleting the directory, the procedure tests if the directory exists by calling g2-directory-exists and signals the error grtl-could-not-delete-directory if it could not delete it.

grtl-format-filename

Synopsis

grtl-format-filename
    (*filename*: text)
    -> *filename*: text

Argument	Description
filename	The filename to format.

Return Value	Description
filename	The resulting filename.

Description

Formats a filename by removing leading and trailing spaces. On Windows platforms, it also converts the filename to uppercase.

grtl-extract-directory

Synopsis

grtl-extract-directory
(*pathname*: text, *extension*: text)
-> *directory*: text

Argument	Description
pathname	The full pathname to a KB file.
extension	The file extension.

Return Value	Description
directory	The resulting directory.

Description

Extracts the directory from a complete pathname that includes the drive and directory name. For example, if the *pathname* is "c:\program files\gensym\g2-2011\example.kb" and the *extension* is "kb", the resulting directory is "c:\program files\gensym\g2-8.3r0".

grtl-extract-filename

Synopsis

grtl-extract-filename
    (*pathname*: text)
    -> *filename*: text

Argument	Description
pathname	The full pathname to a KB file.

Return Value	Description
filename	The resulting filename with the *.kb* extension.

Description

Extracts the KB filename from a complete pathname that includes the drive and directory name. For example, if *pathname* is "c:\program files\gensym\example.kb", the resulting filename is "example.kb".

grtl-extract-filename-with-extension

Synopsis

grtl-extract-filename-with-extension
(*pathname*: text, *extension*: text)
-> *filename*: text

Argument	Description
pathname	The full pathname to a KB file.
extension	The file extension.

Return Value	Description
filename	The resulting filename with the specified extension.

Description

Extracts the KB filename from a complete pathname that includes the drive and directory name. For example, if the *pathname* is "c:\program files\gensym\example.kb" and the *extension* is "kb", the resulting filename is "example.kb".

grtl-filename-to-pathname

Synopsis

grtl-filename-to-pathname
    (*filename*: text)
    -> *pathname*: text, *exists*: truth-value

Argument	Description
filename	The input filename.

Return Value	Description
pathname	The complete pathname.
exists	True if the file already exists.

Description

Formats the filename to a complete pathname. If the filename argument cannot be accessed, it attempts to locate it in the default directory. The procedure returns true if the file exists.

grtl-get-application-directory

Synopsis

grtl-get-application-directory
    ( )
    -> *directory*: text

Return Value	Description
directory	The application directory.

Description

Returns the application directory as specified in the *config.txt* file. This directory might be different from the bundle installation directory to ease the upgrade process as new bundles are installed.

grtl-get-default-directory

Synopsis

grtl-get-default-directory
    ( )
    -> *directory*: text

Return Value	Description
directory	The default directory.

Description

Returns the default directory. The default directory is extracted from the filename-of-basis-kb of the Savings Parameters system table of the top-level module.

grtl-get-installation-directory

Synopsis

grtl-get-installation-directory
    ( )
    -> *directory*: text

Return Value	Description
directory	The installation directory of the bundle.

Description

Returns the installation directory of the bundle.

grtl-get-file-system-delimiter

Synopsis

grtl-get-file-system-delimiter
    ( )
    -> *delimiter*: text

Return Value	Description
delimiter	On UNIX returns "/". On Windows returns "\".

Description

Returns the pathname delimiter for the current file system.

grtl-localize-pathname

Synopsis

grtl-localize-pathname
(*pathname*: text)
-> *pathname*: text

Argument	Description
pathname	The pathname of a file.

Return Value	Description
pathname	The localized file pathname.

Description

Localizes the input pathname to conform to the file system in use. For example, the path "/gensym/tools/g2" would be converted to "\gensym\tools\g2" if the procedure is executed while running on Windows.

You can provide the following symbols at the beginning of the *pathname* argument, which the procedure replaces at runtime with the appropriate value:

- $INSTALLATION-DIRECTORY — The installation directory of the bundle associated with the running application.
- $APPLICATION-ROOT-DIRECTORY — The application directory, which may be different than the installation directory. This directory is typically where user data and KBs are stored and is specified in the *config.txt* file.
- $TOP-LEVEL-MODULE-NAME — The top-level module name.

grtl-localize-pathname-and-create-directory

Synopsis

grtl-localize-pathname-and-create-directory
    (*pathname*: text)
    -> *pathname*: text

Argument	Description
pathname	The pathname of a file.

Return Value	Description
pathname	The localized file pathname.

Description

Localizes the input pathname to conform to the file system in use and creates any required directories, as required, by calling grtl-create-directory.

14

Server Status and Performance Meters

Describes built-in performance meters and procedures to query the status of the server and performance meters.

Introduction

GRTL provides various built-in performance meters and counters that compute:

- Memory usage, size, and availability.
- System performance such as instance creation count, maximum clock tick length, clock time length, and percentage of runtime.
- Scheduler performance of priority lag times.
- The count of objects, Telewindows clients, procedure invocations, raw events, operator messages, and error objects.

It also provides procedures for enabling and disabling performance meters, getting statistics from performance meters, and getting server and module information.

Classes

grtl-performance-counter

Application-specific performance counter. You use application logic to update performance counters.

Class Inheritance Path

grtl-performance-counter, grtl-performance-meter, quantitative-variable, g2-variable, g2-meter-data-service, variable, variable-or-parameter, object, item

grtl-performance-meter

Performance meter for collecting G2 statistics. G2 automatically updates performance meters.

Class Inheritance Path

grtl-performance-meter, quantitative-variable, g2-variable, g2-meter-data-service, variable, variable-or-parameter, object, item

Procedures

grtl-disable-performance-meters

Synopsis

grtl-disable-performance-meters
( )

Description

Disables the G2 performance meters.

grtl-enable-performance-meters

Synopsis

grtl-enable-performance-meters
(*lag-time*: quantity, *update-interval*: integer, *maximum-history-period*: integer, *maximum-history-size*: integer)

Argument	Description
lag-time	The lag time used to update meters.
update-interval	The update interval in seconds of performance meters.
maximum-history-period	The maximum historical values to keep, in units of seconds.
maximum-history-size	The maximum number of historical data points to keep.

Description

Enables the G2 performance meters with the given specification.

grtl-get-loaded-modules-information

Synopsis

grtl-get-loaded-modules-information
(*client*: ui-client-item)
-> *module-info*: sequence

Argument	Description
client	The client, typically a G2 window.

Return Value	Description
module-info	A sequence of structures with information about the modules that are currently loaded.

Description

Returns an alphabetically sorted sequence of structures, where each structure contains information about a module currently loaded in the server. Only modules that include a gfr-version-information-object are reported. Each structure has the following syntax:

structure
(module-name: *symbol*, version: *string*, version-number: *integer*, compatible-version-number: *integer*)

where:

- Module-name is the grf-module-name.
- Version is the gfr-version-description.
- Version-number is the gfr-version-number.
- Compatible-version-number is the gfr-oldest-compatible-version.

grtl-get-performance-meters

Synopsis

grtl-get-performance-meters
( )
-> *meters*: sequence

Return Value	Description
meters	A list of G2 performance meters.

Description

Returns a list of G2 performance meters. The sequence that is returned contains structures. Each structure contains information about one performance meter. The information for each performance counter are described in the attributes of each structure, using the following syntax:

structure
(performance-meter-name: *text*, performance-meter-value: *quantity*, performance-meter-history: *sequence*)

where:

- performance-meter-name is the name of the performance meter.
- performance-meter-value is a snapshot of the value of the performance meter.
- performance-meter-history is the historical values of the performance meter.

The history is returned as a sequence of structures, where each structure contains two attributes, using this syntax:

structure
(history-value: *quantity*, history-collection-time: *quantity*)

where:

- history-value is the value of the performance meter.
- history-collection-time is the timestamp of the value.

The timestamp is in UNIX timestamp format and therefore represents the number of seconds that have elapsed since midnight, January 1, 1970, Greenwich Mean Time.

The list of built-in performance meters and counters can be grouped into several categories:

- Memory performance meters:

  GRTL-MEMORY-USAGE
  GRTL-MEMORY-SIZE
  GRTL-MEMORY-AVAILABLE
  GRTL-REGION-1-USAGE
  GRTL-REGION-1-SIZE
  GRTL-REGION-1-AVAILABLE
  GRTL-REGION-2-USAGE
  GRTL-REGION-2-SIZE
  GRTL-REGION-2-AVAILABLE
  GRTL-REGION-3-USAGE
  GRTL-REGION-3-SIZE
  GRTL-REGION-3-AVAILABLE

- System performance meters:

  GRTL-PERCENT-RUNTIME
  GRTL-CLOCK-TIME-LENGTH
  GRTL-MAXIMUM-CLOCK-TICK-LENGTH
  GRTL-INSTANCE-CREATION-COUNT

- Scheduler performance meters:

  GRTL-PRIORITY-1-SCHEDULER-TIME-LAG
  GRTL-PRIORITY-2-SCHEDULER-TIME-LAG
  GRTL-PRIORITY-3-SCHEDULER-TIME-LAG
  GRTL-PRIORITY-4-SCHEDULER-TIME-LAG
  GRTL-PRIORITY-5-SCHEDULER-TIME-LAG
  GRTL-PRIORITY-6-SCHEDULER-TIME-LAG
  GRTL-PRIORITY-7-SCHEDULER-TIME-LAG
  GRTL-PRIORITY-8-SCHEDULER-TIME-LAG
  GRTL-PRIORITY-9-SCHEDULER-TIME-LAG
  GRTL-PRIORITY-10-SCHEDULER-TIME-LAG

- General performance counters:

  GRTL-COUNT-OF-OBJECTS
  GRTL-COUNT-OF-TELEWINDOWS-CLIENTS
  GRTL-COUNT-OF-PROCEDURE-INVOCATIONS
  GRTL-COUNT-OF-RAW-GEVM-EVENTS
  GRTL-COUNT-OF-GEVM-MESSAGES
  GRTL-COUNT-OF-ERROR-OBJECTS

grtl-get-server-information

Synopsis

grtl-get-server-information
( )
-> *server-info*: text

Return Value	Description
server-info	Server information as a string.

Description

Returns information about the server such as the G2 version, the server host and port name, and the list of loaded modules. See grtl-get-loaded-modules-information for additional information regarding the loaded modules.

15

Text Localization

Describes procedures for localizing text in different languages.

Introduction

GRTL loads one localization resource file per languages at startup. Thus, GRTL relies on a unique resource file rather than managing different resource files for each module. This makes it easier to manage the localized resource files and separate it from the G2 application.

GRTL provides procedures for localizing text by:

- Adding entries to a resource file.

- Loading and saving resource files.
- Localizing message text.
- Merging localization resources from a file.

grtl-add-to-localization-resources

Synopsis

grtl-add-to-localization-resources
    (*language*: symbol, *key*: symbol, *message*: text)

Argument	Description
language	The language.
key	The resource key.
message	The resource text.

Description

Adds an entry to the GRTL resource localization that is loaded at startup.

grtl-get-localization-file

Synopsis

grtl-get-localization-file
    ( )
    -> *filename*: text

Return Value	Description
filename	The localization resource file.

Description

Returns the localization resource file currently specified. Note that any * character is replaced with the current user language at load/save time of the resource file.

grtl-load-localization-resources

Synopsis

grtl-load-localization-resources
    (*add*: truth-value)

Argument	Description
add	True to load the resources, or false to replace the existing resources.

Description

Loads the localization resource file. The path specification of the resources file is defined via a call to grtl-set-localization-file. Note that the character * is replaced with the language. If the *add* argument is false, all resources in the resource are removed before loading the ones in the file. Otherwise, the resources stored in the file are added to the ones already loaded.

grtl-localize-message

Synopsis

grtl-localize-message
(*key*: symbol, *substitutions*: sequence, *language*: symbol, *client*: ui-client-item)
-> *localized-text*: ext

Argument	Description
key	A symbol that is the key for retrieving the localized text.
substitutions	A sequence of up to 10 values to be substituted into the returned text.
language	The language for the returned text.
client	The client, typically a G2 window.

Return Value	Description
localized-text	The localized text.

Description

Gets the localized version of a text, based on a resource group, key, and language, and performs substitutions, if any. This procedure is an improvement on gfr-localize-message in that it does not require the calling procedure to be written for a fixed number of text substitutions.

grtl-localize-message-without-substitutions

Synopsis

grtl-localize-message-without-substitutions
    (*key*: symbol, *client*: ui-client-item)
    -> *localized-text*: ext

Argument	Description
key	A symbol that is the key for retrieving the localized text.
client	The client, typically a G2 window.

Return Value	Description
localized-text	The localized text.

Description

An optimized version of grtl-localize-message when no substitutions are required within the text.

grtl-localize-message-without-substitutions-using-resource-group

Synopsis

grtl-localize-message-without-substitutions-using-resource-group
(*resource-group-name*: symbol, *key*: symbol, *client*: class ui-client-item)
-> *localized-text*: text

Argument	Description
resource-group-name	The name of a GFR text resource group to use for localizing the message.
key	A symbol that is the key for retrieving the localized text.
client	The client, typically a G2 window.

Return Value	Description
localized-text	The localized text.

Description

An optimized version of grtl-localize-message when no substitutions are required within the text given a specific GFR resource group.

grtl-localize-message-for-client

Synopsis

grtl-localize-message-for-client
(*key*: symbol, *substitutions*: sequence, *client*: ui-client-item)
-> *localized-text*: text

Argument	Description
key	A symbol that is the key for retrieving the localized text.
substitutions	A sequence of up to 10 values to be substituted into the returned text.
client	The client, typically a G2 window.

Return Value	Description
localized-text	The localized text.

Description

Gets the localized version of a text and performs substitutions, if any, based on the language of the GRTL resource group defined in the client. This procedure is an improvement on gfr-localize-message in that it does not require the calling procedure to be written for a fixed number of text substitutions.

grtl-merge-localization-resources-from-file

Synopsis

grtl-merge-localization-resources-from-file
    (*filename*: text, *add*: truth-value)

Argument	Description
filename	The resource filename to merge.
add	When true, adds resources from the file; otherwise, replaces existing resources.

Description

Merges localization files into the grtl-localization-resource. The *filename* may contain the wildcard character (*), which is replaced by the supported languages to access different files for different languages. The *filename* can also contain $APPLICATION-ROOT-DIRECTORY or $INSTALLATION-DIRECTORY, which are replaced by the appropriate settings at runtime.

grtl-save-localization-resources

Synopsis

grtl-save-localization-resources
( )

Description

Saves the localization resource file. The path specification of the resources file is defined via a call to grtl-set-localization-file. Note that the character * is replaced with the language.

16

Application Configuration

Describes procedures for retrieving application configuration information from a configuration file.

Introduction

GRTL allows you to configure a G2 application from an external text file, which is loaded at startup. The configuration file contains parameters and values that set default behavior for your application. These procedures allow you to get attribute values from the configuration file. The default configuration file is called *config.txt*

For a general description of the configuration file and how to load it, see System Configuration.

grtl-get-configuration-attribute

Synopsis

grtl-get-configuration-attribute
(*section*: text, *attribute-name*: symbol)
-> *attribute-value*: text

Argument	Description
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Reads and returns the value of a configuration attribute from the configuration file specified in the command-line for starting G2. The following errors may be signalled: grtl-config-file-could-not-be-opened, grtl-config-file-section-not-found, grtl-config-file-section-attribute-not-found, grtl-cfm-failure.

grtl-get-configuration-attribute-or-default

Synopsis

grtl-get-configuration-attribute-or-default
(*section*: text, *attribute-name*: symbol, *default-value*: text)
-> *attribute-value*: text

Argument	Description
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Reads and returns the value of a configuration attribute from the configuration file specified in the command-line for starting G2. If the attribute cannot be located in the file or if the file cannot be accessed, this procedure returns the default value, and no error condition is signalled or reported.

grtl-get-configuration-file-sections

Synopsis

grtl-get-configuration-file-sections
    (*filename*: text)
    -> *sections*: sequence

Argument	Description
filename	The file name containing the configuration file.

Return Value	Description
sections	A sequence of texts.

Description

Returns the sections in the configuration file.

grtl-get-duration-configuration

Synopsis

grtl-get-duration-configuration
(*section*: text, *attribute-name*: symbol, *default-value*: integer)
-> *attribute-value*: inteteger

Argument	Description
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Extracts a duration configuration from the *config.txt* file. The duration specification can use a G2 interval, for example, 2 days and 5 minutes.

grtl-get-duration-configuration-from-configuration-file

Synopsis

grtl-get-duration-configuration-from-configuration-file
(*filename*: text, *section*: text, *attribute-name*: symbol, *default-value*: integer)
-> *attribute-value*: inteteger

Argument	Description
filename	The filename of the custom configuration file.
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Extracts a duration configuration from the specified custom configuration file. The duration specification can use a G2 interval, for example, 2 days and 5 minutes.

grtl-get-integer-configuration

Synopsis

grtl-get-integer-configuration
(*section*: text, *attribute-name*: symbol, *default-value*: integer)
-> *attribute-value*: integer

Argument	Description
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Extracts an integer configuration from the `config.txt` file.

grtl-get-integer-configuration-from-configuration-file

Synopsis

grtl-get-integer-configuration-from-configuration-file
(*filename*: text, *section*: text, *attribute-name*: symbol, *default-value*: integer)
-> *attribute-value*: integer

Argument	Description
filename	The filename of the custom configuration file.
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Extracts an integer configuration from the specified custom configuration file.

grtl-get-item-configuration

Synopsis

grtl-get-item-configuration
    (*target*: class item, *attribute-names*: sequence, *section*: text)

Argument	Description
target	The target item to configure.
attribute-names	A sequence of attribute names, as symbols, to configure from the configuration file.
section	The name of a section in the configuration file, a G2 module name, by convention.

Description

Imports from the `config.txt` file and configures the specified attributes of the target item. The procedure queries the property types, using the grtl-get-property-type-info method, to extract the values from the configuration file.

grtl-get-item-configuration-from-configuration-file

Synopsis

grtl-get-item-configuration-from-configuration-file
(*filename*: text, *target*: class item, *attribute-names*: sequence, *section*: text)

Argument	Description
filename	The filename of the custom configuration file.
target	The target item to configure.
attribute-names	A sequence of attribute names, as symbols, to configure from the configuration file.
section	The name of a section in the configuration file, a G2 module name, by convention.

Description

Imports from the specified custom configuration file and configures the specified attributes of the target item. The procedure queries the property types, using the grtl-get-property-type-info method, to extract the values from the configuration file.

grtl-get-pathname-configuration

Synopsis

grtl-get-pathname-configuration
(*section*: text, *attribute-name*: symbol, *default-value*: text)
-> *pathname*: text

Argument	Description
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
pathname	The configuration attribute read from the configuration file.

Description

Extracts a pathname configuration from the *config.txt* file, localizes the pathname by calling grtl-localize-pathname, and creates the directory, if required.

grtl-get-pathname-configuration-from-configuration-file

Synopsis

grtl-get-pathname-configuration-from-configuration-file
(*section*: text, *attribute-name*: symbol, *default-value*: text,
create-directory: truth-value)
-> *pathname*: text

Argument	Description
filename	The filename of the custom configuration file.
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.
create-directory	Whether to create the directory.

Return Value	Description
pathname	The configuration attribute read from the configuration file.

Description

Extracts a pathname configuration from the specified custom configuration file, localizes the pathname by calling grtl-localize-pathname, and creates the directory, if required.

grtl-get-quantity-configuration

Synopsis

grtl-get-quantity-configuration
(*section*: text, *attribute-name*: symbol, *default-value*: quantity)
-> *attribute-value*: quantity

Argument	Description
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Extracts a quantity configuration from the `config.txt` file.

grtl-get-quantity-configuration-from-configuration-file

Synopsis

grtl-get-quantity-configuration-from-configuration-file
(*filename*: text, *section*: text, *attribute-name*: symbol, *default-value*: quantity)
-> *attribute-value*: quantity

Argument	Description
filename	The filename of the custom configuration file.
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Extracts a quantity configuration from the specified custom configuration file.

grtl-get-symbol-configuration

Synopsis

grtl-get-symbol-configuration
(*section*: text, *attribute-name*: symbol, *default-value*: symbol)
-> *attribute-value*: symbol

Argument	Description
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Extracts a symbol configuration from the *config.txt* file. If the symbol cannot be located or an empty string is specified in the configuration file, the procedure returns the symbol none.

grtl-get-symbol-configuration-from-configuration-file

Synopsis

grtl-get-symbol-configuration-from-configuration-file
(*filename*: text, *section*: text, *attribute-name*: symbol, *default-value*: symbol)
-> *attribute-value*: symbol

Argument	Description
filename	The filename of the custom configuration file.
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Extracts a symbol configuration from the specified custom configuration file. If the symbol cannot be located or an empty string is specified in the configuration file, the procedure returns the symbol none.

grtl-get-truth-value-configuration

Synopsis

grtl-get-truth-value-configuration
(*section*: text, *attribute-name*: symbol, *default-value*: truth-value)
-> *attribute-value*: truth-value

Argument	Description
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Extracts a truth-value configuration from the *config.txt* file. The procedure returns false by default, except if the default value of the attribute in the configuration file is true.

grtl-get-truth-value-configuration-from-configuration-file

Synopsis

grtl-get-truth-value-configuration-from-configuration-file
(*section*: text, *attribute-name*: symbol, *default-value*: truth-value)
-> *attribute-value*: truth-value

Argument	Description
filename	The filename of the custom configuration file.
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.
attribute-name	The name of a configuration attribute within the specified configuration file section.
default-value	The default value if the configuration attribute cannot be located in the configuration file.

Return Value	Description
attribute-value	The configuration attribute read from the configuration file.

Description

Extracts a truth-value configuration from the specified custom configuration file. The procedure returns false by default, except if the default value of the attribute in the configuration file is true.

grtl-save-item-configuration-from-configuration-file

Synopsis

grtl-save-item-configuration-from-configuration-file
(*new-filename*: text, *backup-filename*: text, *target*: class item, *attribute-names*: sequence, *section*: text)

Argument	Description
new-filename	The file name to which to save the configuration file.
backup-filename	The file name to rename the previous configuration file.
target	The target item to configure.
attribute-names	A sequence of names of configuration attributes within the specified section of the configuration file to save.
section	The name of a section in the configuration file, which is the name of a G2 module, by convention.

Description

Saves the specified custom configuration file and configures the specified attributes of the target item. The procedure queries the property types, using the grtl-get-property-type-info method, to extract the values from the configuration file.

17

Command Lines

Describes procedures for retrieving options added to the command line that started G2.

Introduction

GRTL provides procedures for getting the value of a command-line option that was used to start G2 and to determine if a command-line option was used.

grtl-command-line-option-specified

Synopsis

grtl-command-line-option-specified
    (*option-name*: text)
    -> *specified*: truth-value

Argument	Description
option-name	The command-line option name.

Return Value	Description
specified	True if the option was specified in command-line, else false.

Description

Given a command-line option name specified in the command-line used to launch G2, this procedure returns a truth-value indicating whether or not the option was specified in the command-line. For example, if G2 is launched with the command-line `g2.exe -ok my-auth.ok -debug`, then calling this procedure with "debug" returns true.

grtl-get-command-line-option

Synopsis

grtl-get-command-line-option
    (*option-name*: text)
    -> *option-value*: text

Argument	Description
option-name	The command-line option name.

Return Value	Description
option-value	The command-line option value from the command line.

Description

Given a command-line option name specified in the command line used to launch G2, this procedure returns the textual representation of the command-line option value that immediately follows the option name. For example, if G2 is launched with the command line `g2.exe -ok my-auth.ok -trace-level 4`, then calling this procedure with "trace-level" returns the text "4". This procedure signals the error grtl-command-line-option-failure if it fails to find the option name specified or if the option has no value.

Part III

User Interface Operations

Chapter 18: General User Management

Describes general user management functions and procedures.

Chapter 19: Connections

Describes procedures to manage G2 connections, including methods that are automatically called when connections are established to validate the user operation.

Chapter 20: User Preferences

Describes classes and procedures to manage user preferences.

Chapter 21: Operator Messages

Describes general user interface procedures to generate operator messages.

Chapter 22: Menus and Toolbars

Describes procedures to manage menus, toolbars, status bars, and keyboard shortcuts.

Chapter 23: Project Menu and Navigator

Describes a class to create and organize menu choices in the Project menu, as well as the structure of the Navigator and the behavior of the Manage dialog.

Chapter 24: Palettes

Describes a class to create and organize palettes, as well as procedures to manipulate the palettes.

Chapter 25: Child Windows and Panes

Describes procedures to manage child windows and panes.

Chapter 26: Dialogs

Describes general procedures related to dialogs, which provide an abstract API that displays a Microsoft dialog or a UIL dialog, depending on the system and the user configuration.

Chapter 27: Audit Trail

Describes how to enable the audit trail.

18

General User Management

Describes general user management functions and procedures.

Introduction

You use the general user management functions and procedures to:

- Determine the current user mode.
- Get the current user mode and user name.
- Add and delete users.
- Change user passwords.
- Validate users.
- Get a list of currently logged in users.
- Show a workspace ensuring it is always visible.

User Management Activities

The user management activities within the life cycle of a user session include:

- Upon connection or user mode change:
  - If necessary, creates a grtl-user-preference for that user and initializes it, based on the active grtl-default-user-preferences.
  - Creates a grtl-user-session.
  - Configures the user interface by calling the procedures specified on the grtl-user-preferences associated with the user. This procedure builds the menu bar and status bar specified in the grtl-user-preferences, if enabled and valid for the current user mode, and instantiates enabled and valid toolbars and child windows for the current user mode.
  - Tracks the user interface in the grtl-user-session associated with Telewindows, including the menu, toolbar or child window handles, menu command information, and so on.
- At runtime, when the user triggers a command via menus in the menu bar, popup menus, toolbars, or keyboard shortcuts, executes the command by calling the procedure named in the menu-command-execute-procedure of the grtl-user-preferences associated with the user, which implements built-in core commands and dispatches to either global commands or to item-specific commands as described in Menu Commands.

Specifying Additional User Modes

You can specify additional user modes beyond the built-in user modes: operator, modeler, developer, system-administrator, and administrator. When defined, the additional user modes appear in the User Mode list and in the Tools > User Mode menu. If the user switches to a non built-in user mode, you can specify a custom callback procedure on the user preference to customize the layout of the user interface.

To support this feature, GRTL provides available-user-modes attribute of the active grtl-module-settings. The user modes specified in the available-user-modes attribute specify the default user modes if none is specified in the *g2.ok* file. If G2 is running in secure mode, the user mode specified in the *g2.ok* file for a particular user takes precedence over the global specification in the available-user-modes of the active grtl-module-settings object.

Functions

grtl-is-in-administrator-mode

Synopsis

grtl-is-in-administrator-mode
    (*client*: ui-client-item)
    -> *result*: truth-value

Argument	Description
client	The current window.

Return Value	Description
result	True if the user is in administrator mode.

Description

Returns true if the user is in administrator, or if the administrative-permission attribute of the user preference is true.

grtl-is-in-developer-mode

Synopsis

grtl-is-in-developer-mode
    (*client*: ui-client-item)
    -> *result*: truth-value

Argument	Description
client	The current window.

Return Value	Description
result	True if the user is in developer, system-administrator, or administrator mode.

Description

Returns true if the user is in developer, administrator, or system-administrator mode.

grtl-is-in-modeler-mode

Synopsis

grtl-is-in-modeler-mode
    (*client*: ui-client-item)
    -> *result*: truth-value

Argument	Description
client	The current window.

Return Value	Description
result	True if the user is in modeler, developer, system-administrator, or administrator mode.

Description

Returns true if the user is in modeler, developer, system-administrator, or administrator mode.

grtl-is-in-operator-mode

Synopsis

grtl-is-in-operator-mode
    (*client*: ui-client-item)
    -> *result*: truth-value

Argument	Description
client	The current window.

Return Value	Description
result	True if the user is in operator mode.

Description

Returns true if the user is in operator mode.

grtl-is-in-system-administrator-mode

Synopsis

grtl-is-in-system-administrator-mode
    (*client*: ui-client-item)
    -> *result*: truth-value

Argument	Description
client	The current window.

Return Value	Description
result	True if the user is in system-administrator or administrator mode.

Description

Returns true if the user is in system-administrator or administrator mode, or if the administrative-permission attribute of the user preference is true.

grtl-get-user-mode

Synopsis

grtl-get-user-mode
    (*client*: ui-client-item)
    -> *user-mode*: symbol

Argument	Description
client	The current window.

Return Value	Description
user-mode	The user mode of the current window.

Description

Returns the current user mode for specified client window, as a symbol.

grtl-get-user-name

Synopsis

grtl-get-user-name
    (*client*: ui-client-item)
    -> *user-name*: symbol

Argument	Description
client	The current window.

Return Value	Description
user-name	The user name of the current window.

Description

Returns the user name, as a symbol, for the specified client window. If the user name of the client window is not defined, the function returns the user name in the operating system, as symbol.

Procedures

grtl-add-user

Synopsis

grtl-add-user
(*user-name*: text, *password*: text, *password-expiration*: truth-value,
password-expiration-date: integer, *password-expiration-month*: integer,
password-expiration-year: integer, *is-developer*: truth-value,
is-modeler: truth-value, *is-operator*: truth-value)

Argument	Description
user-name	The user name of the new user.
password	The password of the new user.
password-expiration	Whether the password expires.
password-expiration-date	The password expiration date.
password-expiration-month	The password expiration month.
password-expiration-year	The password expiration year.
is-developer	Whether the user is allowd access in developer mode.
is-modeler	Whether the user is allowed access in modeler mode.
is-operator	Whether the user is allowed access in operator mode.

Description

Adds a new user to the application in a given user mode and updates the `g2.ok` file. Note that for security reasons, you cannot use this procedure to add a user in administrator or system-administrator mode. You have to add users through the user interface or edit the OK file.

grtl-change-user-password

Synopsis

grtl-change-user-password
(*user-name*: text, *old-password*: text, *new-password*: text)

Argument	Description
user-name	The user name of the user whose password to change.
old-password	The old password.
new-password	The new password.

Description

Updates the password of the specified user and updates the *g2.ok* file.

grtl-delete-user

Synopsis

grtl-delete-user
    (*user-name*: text)

Argument	Description
user-name	The user name of the user to delete.

Description

Deletes a user from the application and removes the user from the *g2.ok* file. If the user is connected, the user is not logged out but will not be able to log in again after logging out.

grtl-get-remote-users

Synopsis

grtl-get-remote-users
    ( )
    -> *users*: sequence

Return Value	Description
users	A sequence of currently connected user names.

Description

Returns a sequence of currently connected users, as texts.

grtl-show-workspace

Synopsis

grtl-show-workspace
(*workspace*: class kb-workspace, *window*: class g2-window)

Argument	Description
workspace	The kb-workspace to show.
window	The current window.

Description

Displays a workspace, ensuring that the workspace is always at least partially visible, independent of screen resolution.

The G2 expression show *workspace* on *window* does not ensure that the workspace is always visible in the screen; the workspace may not be in the visible area of the screen. This procedure calls g2-ui-show-workspace to ensure the workspace is visible in the screen. It only shows the workspace if the specified window is connected; for example, it does not show the workspace in the gfr-default-window.

grtl-validate-user

Synopsis

grtl-validate-user
    (*user-name*: symbol, *win*: class g2-window)
    -> *valid*: truth-value, *error*: text

Argument	Description
user-name	The user name of the user to validate.
win	The current window.

Return Value	Description
valid	True if the user is valid, false otherwise.
error	Error text if the user is not valid.

Description

Returns true if the user name is valid and a text describing the error message, if any.

19

Connections

Describes procedures to manage G2 connections, including methods that are automatically called when connections are established to validate the user operation.

Introduction

GRTL provides methods and procedures for managing G2 connections, including:

- Validating connections, junctions, connected items, and stub deletion.
- Enabling connection validation.

- Getting the input and output items of a connection.
- Creating connection stubs.
- Deleting connections and stubs.
- Getting the workspace of a connection.

connection::grtl-validate-joined-connection

Synopsis

connection::grtl-validate-joined-connection
    (*cxn*: connection, *client*: ui-client-item)
    -> *valid*: truth-value

Argument	Description
cxn	The connection to which another connection has been added.
client	The client, typically a G2 window.

Return Value	Description
valid	True if the connection is valid, false otherwise.

Description

If connection validation has been enabled via a call to grtl-enable-connection-validation, this method is called whenever two connections are joined. This method should validate the connection and, if valid, return true; otherwise, it should return false.

connection::grtl-validate-junction

Synopsis

connection::grtl-validate-junction
    (*cxn*: connection, *client*: ui-client-item)
    -> *valid*: truth-value

Argument	Description
cxn	The connection to which a junction has been added.
client	The client, typically a G2 window.

Return Value	Description
valid	True if the junction is valid, false otherwise.

Description

If connection validation has been enabled via a call to grtl-enable-connection-validation, this method is called whenever a junction block is added to a connection. This method should validate that a junction block can be added and, if so, return true; otherwise, it should return false.

item::grtl-validate-connection

Synopsis

item::grtl-validate-connection
(*itm*: item, *cxn*: connection, *client*: ui-client-item, *type*: symbol)
-> *valid*: truth-value

Argument	Description
itm	The item to which the connection is connected.
cxn	The connection to validate.
client	The client, typically a G2 window.
type	The type of connection, which should be connection-source or connection-target.

Return Value	Description
valid	True if the connection is valid, false otherwise.

Description

If connection validation has been enabled via a call to grtl-enable-connection-validation, this method is called whenever a connection to an item should be validated. This method should determine if the connection is valid and if so return true. Otherwise it should return false, which will result in the connection been deleted.

item::grtl-validate-stub-deletion

Synopsis

item::grtl-validate-stub-deletion
    (*itm*: item, *client*: ui-client-item)

Argument	Description
itm	The item for which a stub has been removed.
client	The client, typically a G2 window.

Description

If connection validation has been enabled via a call to grtl-enable-connection-validation, this method is called whenever a stub is deleted. This method can validate that the stub can be removed and, if not, can re-created the missing stubs.

grtl-connection-input-and-output

Synopsis

grtl-connection-input-and-output
    (*cxn*: connection)
    -> *input*: item-or-value, *output*: item-or-value

Argument	Description
cxn	The connection for which to return the items connected at the input and output.

Return Value	Description
input	The input object.
output	The output object.

Description

Returns the input and output items for a connection. If no input or output is connected to the connection, this procedure returns false instead of the connected object.

grtl-create-new-stub

Synopsis

grtl-create-new-stub
(*obj*: object, *connection-class*: symbol, *side*: symbol, *position*: integer, *portname*: symbol, *direction*: symbol)

Argument	Description
obj	The object to which a stub should be added.
connection-class	The class of the connection stub.
side	The side of the connection stub, which should be top, bottom, left, or right.
position	The position of the connection stub along the side.
portname	The name of the associated port.
direction	The direction of the connection stub, which should be input or output.

Description

Adds a new orthogonal stub to an object.

grtl-delete-connection-removing-illegal-stub

Synopsis

grtl-delete-connection-removing-illegal-stub
(*object*: object, *cxn*: connection)

Argument	Description
object	The object from which to remove the connection.
cxn	The connection to remove.

Description

Removes the connection from an object, including any related stubs.

grtl-enable-connection-validation

Synopsis

grtl-enable-connection-validation
    ( )
    -> *enabled*: truth-value

Return Value	Description
enabled	True if enabled, or false if connection validation could not be enabled.

Description

Enables connection validation callbacks to be called. If enabled, the methods grtl-validate-connection, grtl-validate-stub-deletion, grtl-validate-junction, and grtl-validate-joined-connection are automatically called when the connection or junction is created. You provide the validation logic in the method.

grtl-get-workspace-of-connection

Synopsis

grtl-get-workspace-of-connection
    (*cxn*: connection)
    -> *workspace*: item-or-value

Argument	Description
cxn	The connection.

Return Value	Description
workspace	The workspace or false.

Description

Returns the workspace of a connection or false if the workspace does not exist.

20

User Preferences

Describes classes and procedures to manage user preferences.

Introduction

User preferences are used to track user specific configurations, such as the default user mode to be used upon connection and the user's email address. Each user can configure some of his/her user preferences, while a system administrator can configure additional user-specific configurations. Once a user preference has been created, either explicitly or implicitly, it is never deleted. A user preference object can be associated with one or more g2-window instances.

A user session stores configuration items specific to the current Telewindows session, such as window or menu handles. A user session object is associated with

one g2-window instance. The user session object is deleted upon disconnecting from Telewindows.

Creating User Preferences and User Sessions

One user preference object is created per user name, based on the g2-user-name of the g2-window. Upon connection to the server, via Telewindows, for example, the grtl-get-user-preferences procedure is called to configure the user preference, which:

1. Creates a grtl-user-session, if none is already associated with the g2-window.
2. Searches for a grtl-user-preference associated with the g2-window. If it exists, it is returned. If none exists, it searches for an existing user preference for that user, based on the g2-user-name, as follows:
   a. If one exists, it associates the user preference with the g2-window, calls the grtl-initialize method for the user preference, for example, and associates the user preference with the g2-window.
   b. If none exists, it creates a new user preference, associates the user preference with the g2-window, and calls the grtl-initialize method for the session, associates the user preference with the g2-window. The user-configuration-class-name attribute of the active grtl-module-settings instance specifies the class of the user preference to create. If GRTL is loaded, the default value is grtl-user-preference, and if GEVM is loaded, the default value is gevm-user-preference.

Note that the default user preference can also be imported from a file. For details, see Importing and Exporting User Preferences.

Also note that upon connection, that is, the first time a grtl-user-preference and grtl-user-session are associated with a g2-window, grtl-get-user-preferences also ensures that the Telewindows connection is supported. If it is not supported, the connection is dropped. To validate a connection type, the g2-window-ui-style of the g2-window needs to be a member of the application-valid-window-ui-styles of the active grtl-module-settings instance. This enables you to support only certain window styles, such as multi-window.

Initializing User Preferences

User preferences are initialized by calling the grtl-initialize method. This method initializes the attributes of the user preference with values from the active grtl-default-user-preferences instance. The active grtl-default-user-preferences instance is the one that is the highest in the module hierarchy. Therefore, you can create your own configuration by cloning the currently active user preference, storing it in a higher level module, and updating the configuration, as appropriate.

Some attributes of the user preference are only initialized with the values of the active grtl-default-user-preference if the *InitializeWithPreference* argument of this method is true. This ensures that GRTL maintains user-specific configurations across Telewindows connections. The attributes are:

- email-address
- mobile-email-address
- default-user-mode
- home-location
- location-history-size
- indicate-items-upon-menu-selection
- disconnect-permission
- shutdown-permission
- acknowledge-messages-permission
- delete-messages-permission
- use-g2-logbook
- beep-enabled
- show-message-browser-by-default
- acknowledge-messages-upon-selection
- tabbed-mdi-mode
- ui-theme
- show-native-user-interface-preference

Importing and Exporting User Preferences

The grtl-module-settings object defines the user-preferences-filename attribute to specify the file name used for importing and exporting the user configuration.

To import and export user preferences from and to a file:

1. Configure the user-preferences-filename attribute in the grtl-module-settings object.
2. Switch to System-Administrator mode.
3. Choose Tools > Import User Preferences or Tools > Export User Preferences.

Each time the user preferences are saved, the existing user preference file is renamed to the same name with the *.bak* extension. The default location and file name is *APPLICATION-ROOT-DIRECTORY/g2i/data/user-preferences.txt*.

When importing user preferences from the file, GRTL creates a user preference for each user preference that does not already exist. If a user preference already exists, GRTL updates its settings based on the contents of the file. The contents of the user preferences file is imported upon reset.

Note that if a secure G2 is used, the users are not automatically added to the *g2.ok* file; you must add them manually.

Here is a the format of the user preferences file with sample values:

```
[myUserName]
EMAIL-ADDRESS=
MOBILE-EMAIL-ADDRESS=
DEFAULT-USER-MODE=modeler
HOME-LOCATION=default view
INDICATE-ITEMS-UPON-MENU-SELECTION=false
DISCONNECT-PERMISSION=true
SHUTDOWN-PERMISSION=true
ACKNOWLEDGE-MESSAGES-PERMISSION=true
DELETE-MESSAGES-PERMISSION=true
CONFIGURATION-PERMISSION=true
USE-G2-LOGBOOK=false
BEEP-ENABLED=true
SHOW-EXTENDED-MENUS=true
SHOW-MESSAGE-BROWSER-BY-DEFAULT=true
ACKNOWLEDGE-MESSAGES-UPON-SELECTION=false
TABBED-MDI-MODE=false
UI-THEME=WINDOWS-THEME-2003
MENUBAR-NAME=DEFAULT-MENUBAR
STATUSBAR-NAME=DEFAULT-STATUS-BAR
RESTORE-LAST-CHILD-PANE-SETTINGS=false
EMAIL-NOTIFICATION=NEVER
MOBILE-EMAIL-NOTIFICATION=NEVER
ENABLE-STATUS-BAR-MESSAGE-BROWSER=true
MODELER-MESSAGE-BROWSER-TEMPLATE=GEVM-MODELER-MSG-VIEW-TEMPLATE
OPERATOR-MESSAGE-BROWSER-TEMPLATE=GEVM-OPERATOR-MSG-VIEW-TEMPLATE
TELNET-SHELL-COMMAND="C:\Program Files\PuTTY\putty.exe"
```

Default User Mode

Upon connection, if the grtl-set-default-user-mode-for-new-users logical parameter is true, the default user mode for the g2-window is set to the user mode specified by the default-user-mode attribute of the user's grtl-user-preference. This enables you to specify different initial user modes per user. This is done in grtl-get-user-preferences when a user preference is initially assigned to the g2-window.

Getting User Preferences and User Sessions

To retrieve the user preference for a g2-window, call grtl-get-user-preferences, which returns the user preference and the user session objects.

Extending User Preferences

You can extend the user preference object and add your custom attributes. To do this:

1. Create a subclass of grtl-user-preferences, or, if you have merged in GEVM, create a subclass of gevm-user-preferences.
2. Create an instance of a grtl-module-settings by copying it from the currently active module settings, and store it in the KB where you created your user preference subclass or any KB module higher in the module hierarchy.
3. Change the user-configuration-class-name attribute of your newly created grtl-module-settings to the name of your class created in step 1.
4. Restart G2.

Classes

grtl-user-preferences

A user preference is unique per user name and contains user-specific preferences. When users connect to the server via Telewindows, the user name of the g2-window is used to locate an existing grtl-user-preference, If none exists, a new instance is created and initialized with the active grtl-default-user-preference. When the user disconnects, the instance is kept and reused the next time the user connects to the server. The specific instance that is created upon connection is specified in the grtl-module-settings and, therefore, enables applications to extend the grtl-user-preferences class with application-specific settings.

The following logic is used to determine the user name used to locate the grtl-user-preference given the ui-client-item:

```
if ui-client-item is a g2-window and the g2-user-name of ui-client-item exists then
    the g2-user-name of ui-client-item else
    if ui-client-item is a g2-window and
        the g2-window-user-name-in-operating-system of ui-client-item exists then
            text-to-symbol (upper-case-text
                (the g2-window-user-name-in-operating-system of ui-client-item)) else
                if ui-client-item is a ui-client-session and
                    the ui-client-session-user-name of ui-client-item exists then
                    the ui-client-session-user-name of ui-client-item else
                    if ui-client-item is a ui-client-session and
                        the ui-client-session-user-name-in-operating-system of
                            ui-client-item exists then
                        text-to-symbol (upper-case-text
                            (the ui-client-session-user-name-in-operating-system of
                            ui-client-item)) else
                            the symbol unknown
```

Session-specific values for each Telewindows connection are tracked in instances of grtl-user-session.

Class Inheritance Path

grtl-user-preferences, grtl-object-with-key, object, grtl-item-with-key, grtl-item, item

Attributes

Attribute	Description
user-name	See grtl-default-user-preferences.
Allowable values:	Any symbol
Default value:	anonymous

email-address See grtl-default-user-preferences.

Allowable values: Any text

Default value: ""

mobile-email-address See grtl-default-user-preferences.

Allowable values: Any text

Default value: ""

default-user-mode See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: operator

home-location See grtl-default-user-preferences.

Allowable values: Any text

Default value: "default"

location-history See grtl-default-user-preferences.

Allowable values: Any sequence

Default value: sequence ()

location-history-index See grtl-default-user-preferences.

Allowable values: Any integer

Default value: -1

location-history-size See grtl-default-user-preferences.

Allowable values: Any integer

Default value: 20

indicate-items-upon-menu-selection See grtl-default-user-preferences.

Allowable values: Any truth-value

Default value: false

disconnect-permission See grtl-default-user-preferences.

Allowable values: Any truth-value

Default value: false

shutdown-permission See grtl-default-user-preferences.

Allowable values: Any truth-value

Default value: true

configuration-permission See grtl-default-user-preferences.

Allowable values: Any truth-value

Default value: true

acknowledge-messages-permission See grtl-default-user-preferences.

Allowable values: Any truth-value

Default value: true

delete-messages-permission See grtl-default-user-preferences.

Allowable values: Any truth-value

Default value: true

use-g2-logbook See grtl-default-user-preferences.

Allowable values: Any truth-value

Default value: false

beep-enabled See grtl-default-user-preferences.

Allowable values: Any truth-value

Default value: true

show-extended-menus If true, displays extended menus in the project menu bar listing instances.

Allowable values: Any truth-value

Default value: true

show-native-user-interface-preference See grtl-default-user-preferences.

Allowable values: Any truth-value

Default value: false

show-message-browser-by-default See grtl-default-user-preferences.

Allowable values: Any truth-value

Default value: true

acknowledge-messages-upon-selection See grtl-default-user-preferences.

Allowable values: Any truth-value

Default value: false

ui-theme The user interface theme to use, as a symbol, which changes the look of all connected Telewindows sessions for that user.

Allowable values: windows-theme-2000, windows-theme-2001, windows-theme-2002, or windows-theme-2003

Default value: windows-theme-2003

release-user-interface-procedure See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: grtl-release-user

show-custom-user-interface-procedure See grtl-default-user-preferences.

The value is initialized from the active grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

show-operator-user-interface-procedure See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

show-modeler-user-interface-procedure See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

show-developer-user-interface-procedure

See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

show-system-administrator-user-interface-procedure

See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

resize-custom-user-interface-procedure

See grtl-default-user-preferences.

The value is initialized from the active grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

resize-operator-user-interface-procedure

See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

resize-modeler-user-interface-procedure

See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

resize-developer-user-interface-procedure

See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

resize-system-administrator-user-interface-procedure See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

refresh-user-interface-procedure See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

view-item-procedure See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

view-metrics-procedure See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

view-process-map-procedure See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

view-equipment-procedure See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

view-event-detection-procedure See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

view-fault-models-procedure See grtl-default-user-preferences.

Allowable values: Any symbol

Default value: unspecified

user-attribute-1 Application-specific attribute for any purpose, which means you do not need to subclass the user preference to provide application-specific behavior.

Allowable values: Any item or value

Default value: none

user-attribute-2 Application-specific attribute for any purpose, which means you do not need to subclass the user preference to provide application-specific behavior.

Allowable values: Any item or value

Default value: none

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

menubar-name The default menu bar to display for the user.

Allowable values: symbol

Default value: default-menubar

menu-command-execute-procedure A procedure to be called when executing menu actions triggered from the menubar, popup menus, toolbars, or keyboard accelerators. Menu commands are uniquely identified by the symbolic name of the command.

Allowable values: symbol

Default value: none

statusbar-name The default status bar to display for the user.

Allowable values: symbol

Default value: default-status-bar

relation-display-mode The display mode when displaying related objects, using the relation browser utility.

Allowable values: symbol

Default value: none

zooming-factor The last custom zooming factor the user selected.

Allowable values: quantity

Default value: 0

tabbed-mdi-mode See grtl-default-user-preferences.

Allowable values: truth-value

Default value: false

restore-last-child-pane-settings Whether to restore the last size and position of the pane when the user displays it (true) or whether to use the default position and size of the pane (false).

Allowable values: truth-value

Default value: true

set-default-user-mode-upon-connection When true, the user mode of the g2-window is overwritten by the default-user-mode attribute of the user's grtl-user-preferences object when the user connects. Otherwise, the default KB user mode or the user mode specified as a command-line option is used. This enables you to specify the default behavior for all users in the default user preference and overwrite it for selective users.

Allowable values: truth-value

Default value: false

Methods

grtl-user-preferences::grtl-show-properties

grtl-default-user-preferences

Provides a default user preference that is active for a particular application. The active user preference is determined at reset time and is the highest user preference instance in the G2 module hierarchy. At run time and when a user connects to G2 via Telewindows, for example, the logic determines if an existing grtl-user-preferences exists, based on the user name in the G2 window. If none exists, GRTL creates one and initializes it with the content of the active default gfr-module-setting.

Class Inheritance Path

grtl-default-user-preferences, gfr-module-setting, object, item

Attributes

Attribute	Description
email-address	The default email address.
Allowable values:	Any text
Default value:	""
mobile-email-address	The default mobile email address.
Allowable values:	Any text
Default value:	""
user-mode	The default user mode.
Allowable values:	Any symbol
Default value:	operator
home-location	The default workspace to display.
Allowable values:	Any text
Default value:	"default view"
location-history-size	The maximum number of user interface events to keep in the history.
Allowable values:	Any integer
Default value:	20

gqs-update-latency The default interval for updating message queues.

Allowable values: Any float

Default value: 1.0

gevm-max-events-to-queue The maximum number of events to keep in the GEVM message queue.

Allowable values: Any integer

Default value: 100000

message-filter The default filter for filtering messages.

Allowable values: Any grtl-user-message-filter

Default value: grtl-user-message-filter

indicate-items-upon-menu-selection Whether to go to the item instance when creating the item from a menu (true) or whether to display the properties dialog or detail workspace directly (false).

Allowable values: Any truth-value

Default value: false

Notes: See Configuration File.

disconnect-permission Whether to allow the user to disconnect the client from the server.

Allowable values: Any truth-value

Default value: false

shutdown-permission Whether to allow the user to shutdown the server.

Allowable values: Any truth-value

Default value: true

configuration-permission Whether to allow the user the ability to configure applications, as well as run them as operators.

Allowable values: Any truth-value

Default value: true

acknowledge-messages-permission Whether to allow the user to acknowledge messages in a message browser.

Allowable values: Any truth-value

Default value: true

delete-messages-permission Whether to allow the user to delete messages in a message browser.

Allowable values: Any truth-value

Default value: true

use-g2-logbook Whether to display the G2 logbook.

Allowable values: Any truth-value

Default value: false

beep-enabled Whether to enable beeping.

Allowable values: Any truth-value

Default value: true

show-native-user-interface Set to false to display the G2 classic user interface instead of the native user interface. Note that this attribute might be deprecated in a future release.

Allowable values: Any truth-value

Default value: true

show-message-browser-by-default Whether to show the message browser by default or the home-location workspace when switching to operator mode.

Allowable values: Any truth-value

Default value: true

acknowledge-messages-upon-selection Whether messages are automatically acknowledged upon selection in a message browser.

Allowable values: Any truth-value

Default value: false

release-user-interface-procedure A procedure to execute when the user interface associated with the user preference is released.

Allowable values: Any symbol

Default value: grtl-release-user

show-custom-user-interface-procedure A procedure to execute when the user switches to a custom user mode.

Allowable values: Any symbol

Default value: unspecified

show-operator-user-interface-procedure A procedure to execute when the user switches to operator mode.

Allowable values: Any symbol

Default value: grtl-show-user-interface-default-handler

show-modeler-user-interface-procedure A procedure to execute when the user switches to modeler mode.

Allowable values: Any symbol

Default value: grtl-show-user-interface-default-handler

show-developer-user-interface-procedure A procedure to execute when the user switches to developer mode.

Allowable values: Any symbol

Default value: grtl-show-user-interface-default-handler

show-system-administrator-user-interface-procedure A procedure to execute when the user switches to system-administrator mode.

Allowable values: Any symbol

Default value: grtl-show-user-interface-default-handler

resize-custom-user-interface-procedure A procedure to execute when the user interface associated with the user preference is resized in a custom user mode.

Allowable values: Any symbol

Default value: unspecified

resize-operator-user-interface-procedure A procedure to execute when the user interface associated with the user preference is resized in operator mode.

Allowable values: Any symbol

Default value: unspecified

resize-modeler-user-interface-procedure A procedure to execute when the user interface associated with the user preference is resized in modeler mode.

Allowable values: Any symbol

Default value: unspecified

resize-developer-user-interface-procedure A procedure to execute when the user interface associated with the user preference is resized in developer mode.

Allowable values: Any symbol

Default value: unspecified

resize-system-administrator-user-interface-procedure A procedure to execute when the user interface associated with the user preference is resized in system-administrator mode.

Allowable values: Any symbol

Default value: unspecified

refresh-user-interface-procedure A procedure to execute when the user interface associated with the user preference is refreshed.

Allowable values: Any symbol

Default value: unspecified

view-item-procedure A procedure to execute when pointing to an item such as when selecting show details.

Allowable values: Any symbol

Default value: unspecified

view-metrics-procedure A procedure to execute for viewing event metrics from the Operator console.

Allowable values: Any symbol

Default value: unspecified

view-process-map-procedure — A procedure to execute for viewing the process map from the Operator console.

Allowable values: Any symbol

Default value: unspecified

view-equipment-procedure — A procedure to execute for viewing a domain object from the Operator console.

Allowable values: Any symbol

Default value: unspecified

view-event-detection-procedure — A procedure to execute for viewing GEDP event-detection models from the Operator console.

Allowable values: Any symbol

Default value: unspecified

view-fault-models-procedure — A procedure to execute for viewing SymCure fault models from the Operator console.

Allowable values: Any symbol

Default value: unspecified

ui-theme — The user interface theme to use, as a symbol, which changes the look of all connected Telewindows sessions for that user.

Allowable values: windows-theme-2000, windows-theme-2001, windows-theme-2002, or windows-theme-2003

Default value: windows-theme-2003

menubar — The menubar-name of the menu bar to display for the user.

Allowable values: symbol

Default value: default-menubar

statusbar-name — The statusbar-name of the status bar to display for the user.

Allowable values: symbol

Default value: default-status-bar

tabbed-mdi-mode Whether to use a Multiple Document Interface (MDI) display or a Tabbed Document Interface (TDI) display. The default is to use an MDI display.

Allowable values: truth-value

set-default-user-mode-upon-connection When true, the user mode of the g2-window is overwritten by the default-user-mode attribute of the user's grtl-user-preferences object when the user connects. Otherwise, the default KB user mode or the user mode specified as a command-line option is used. This enables you to specify the default behavior for all users in the default user preference and overwrite it for selective users.

Allowable values: truth-value

Default value: false

Methods

grtl-default-user-preferences::grtl-show-properties

grtl-user-message-filter

A filter specification for filtering messages in a message browser.

Class Inheritance Path

grtl-user-message-filter, object, item

Attributes

Attribute	Description
select-priority-1-messages	Whether to include messages with priority 1.
Allowable values:	Any truth-value
Default value:	true
select-priority-2-messages	Whether to include messages with priority 2.
Allowable values:	Any truth-value
Default value:	true
select-priority-3-messages	Whether to include messages with priority 3.
Allowable values:	Any truth-value
Default value:	true
select-priority-4-messages	Whether to include messages with priority 4.
Allowable values:	Any truth-value
Default value:	true
select-priority-5-messages	Whether to include messages with priority 5.
Allowable values:	Any truth-value
Default value:	true
select-priority-6-messages	Whether to include messages with priority 6.
Allowable values:	Any truth-value
Default value:	true

select-priority-7-messages Whether to include messages with priority 7.

Allowable values: Any truth-value

Default value: true

select-priority-8-messages Whether to include messages with priority 8.

Allowable values: Any truth-value

Default value: true

select-all-priority-messages Whether to include messages with any priority.

Allowable values: Any truth-value

Default value: true

enable-filtering-by-type Whether to allow filtering based on the message type.

Allowable values: Any truth-value

Default value: false

selected-type The message type to use as filter criteria.

Allowable values: Any symbol

Default value: none

enable-filtering-for-target Whether to allow filtering based on the message target.

Allowable values: Any truth-value

Default value: false

selected-target The message target to use as filter criteria.

Allowable values: Any text

Default value: ""

enable-filtering-by-process-map Whether to allow filtering based on the process map.

Allowable values: Any truth-value

Default value: false

selected-process-map The process map to use as filter criteria.

Allowable values: Any text

Default value: ""

enable-filtering-for-category Whether to allow filtering based on the message category.

Allowable values: Any truth-value

Default value: false

selected-category The message category to use as filter criteria.

Allowable values: Any text

Default value: ""

enable-filtering-assigned-to-user Whether to allow filtering based on the assigned user a message.

Allowable values: Any truth-value

Default value: false

selected-user The user name of the assigned user to use as filter criteria.

Allowable values: Any text

Default value: ""

enable-filtering-by-update-time Whether to allow filtering based on the update frequency for a message.

Allowable values: Any truth-value

Default value: false

selected-message-aging The update time to use as filter criteria.

Allowable values: Any item or value

Default value: 0

select-unacknowledged-messages-only Whether to allow filtering based on whether a message has been acknowledged.

Allowable values: Any truth-value

Default value: false

hide-subsumed-messages Whether to hide submessages of a message.

Allowable values: Any truth-value

Default value: false

message-queue-subscription A sequence of message queues to which to subscribe to messages.

Allowable values: Any sequence

Default value: sequence ("Messages", "Alarms", "Root Causes", "Test Actions", "Repair Actions")

custom-message-filter-procedure A custom procedure to execute when the message filter is applied.

Allowable values: Any symbol

Default value: unspecified

grtl-user-session

An internal class that is to be used by advanced users only. The attributes should not be changed.

Class Inheritance Path

grtl-user-session, object, item

Attributes

Attribute	Description
_user-name	The user name of the current user.
Allowable values:	symbol
Default value:	none
_user-mode	The user mode of the current user.
Allowable values:	symbol
Default value:	none
_previous-user-mode	The previous user mode of the current user.
Allowable values:	symbol
Default value:	none
_show-native-user-interface	Whether to show the native user interface.
Allowable values:	truth-value
Default value:	false
_window-width	The width of the window, in pixels.
Allowable values:	integer
Default value:	0
_window-height	The height of the window, in pixels.
Allowable values:	integer

Default value: 0

_selection-callback-handle The handle of the G2 selection callback for this window.

Allowable values: integer

Default value: -1

_menubar The current menu bar instance of the current user.

Allowable values: an instance of a grtl-menu-or-toolbar-instance

Default value: an instance of a grtl-menu-or-toolbar-instance

_popup-menu The current popup menu instance of the current user.

Allowable values: an instance of a grtl-menu-or-toolbar-instance

Default value: an instance of a grtl-menu-or-toolbar-instance

_toolbars A list of grtl-menu-or-toolbar-instance objects, one per available toolbar within the associated Telewindows session.

Allowable values: an instance of an item-list

Default value: an instance of a grtl-menu-or-toolbar-instance

_child-windows A hash table of child window names, as symbols, (the key) and the associated G2 handle of the pane window.

Allowable values: an instance of a hash-table

Default value: an instance of a hash-table

_handle-hash-table A general hash table to keep track of handles, as desired.

Allowable values: an instance of a hash-table

Default value: an instance of a hash-table

_show-status-bar Whether to show the status bar.

Allowable values: truth-value

Default value: true

_message-browser-showing

Allowable values: truth-value

Default value: true

_items-to-delete-upon-disconnect

A list of objects to delete upon disconnect to avoid memory leaks.

Allowable values: sequence

Default value: sequence()

grtl-menu-or-toolbar-instance

An internal class used to track instances of the menubar, popup, and toolbars for a specific Telewindows session. The attributes should not be modified

Class Inheritance Path

grtl-menu-or-toolbar-instance, object, item

Attributes

Attribute	Description
menu-or-toolbar-name	The name of the menu or toolbar.
Allowable values:	symbol
Default value:	g2
menu-or-toolbar-handle	The G2 handle for the menu or toolbar, which is required for low level changes using the G2 APIs.
Allowable values:	integer
Default value:	0
target	The target associated with the popup menu.
Allowable values:	sequence
Default value:	sequence( )
command-handles	A hash table of command names, as symbols, (the key) and associated command information, including the handle of the associated G2 menu.
Allowable values:	an instance of a hash-table
Default value:	an instance of a hash-table
_last-update-time	The last time the menus were updated, enabled, checked, based on the current selection, which is used to collapse changes in batch intervals for efficiency reasons.
Allowable values:	quantity
Default value:	0.0

_update-procedure Procedure called to update the enabled and checked state of menus, based on the current selection.

Allowable values: symbol

Default value: unspecified

Methods and Procedures

Methods

grtl-user-preferences::grtl-show-properties
grtl-default-user-preferences::grtl-show-properties

Procedures

grtl-get-user-preferences
grtl-get-user-preferences-by-name

Function

grtl-use-native-user-interface

grtl-user-preferences::grtl-show-properties

Synopsis

grtl-user-preferences::grtl-show-properties
(*user-preference*: grtl-user-preferences, *client*: ui-client-item)
-> *applied*: truth-value

Argument	Description
user-preference	The user preference whose properties to show.
client	The client, typically a G2 window.

Return Value	Description
applied	Whether the property values were applied to the user preference.

Description

Displays the properties dialog of the specified user preference. This method returns true if the OK button of the dialog is pressed and changes are applied; otherwise, it returns false if no changes were applied.

grtl-default-user-preferences::grtl-show-properties

Synopsis

grtl-default-user-preferences::grtl-show-properties
(*itm*: grtl-default-user-preferences, *client*: ui-client-item)
-> *applied:* truth-value

Argument	Description
itm	The default user preference to view.
client	The client, typically a G2 window.

Return Value	Description
applied	Whether the property values were applied to the user preference.

Description

Displays the properties dialog of the specified default user preference. This method returns true if the OK button of the dialog is pressed and changes are applied; otherwise, it returns false if no changes were applied.

grtl-get-user-preferences

Synopsis

grtl-get-user-preferences
    (*client*: ui-client-item)
    -> *user-preference*: class grtl-user-preferences,
    user-session: class grtl-user-session

Argument	Description
client	The client, typically a G2 window.

Return Value	Description
user-preference	The active user preference.
user-session	The user session for the connection.

Description

Returns the active user preference and user session.

grtl-get-user-preferences-by-name

Synopsis

grtl-get-user-preferences-by-name
    (*user-name*: symbol)
    -> *user-preference*: grtl-user-preferences

Argument	Description
user-name	The user name whose preference to get.

Return Value	Description
user-preference	The user preference associated with the user name.

Description

Returns the user preference associated with a given user name.

grtl-use-native-user-interface

Synopsis

grtl-use-native-user-interface
    (*win*: class g2-window)
    -> *return-value*: truth-value

Argument	Description
win	

Return Value	Description
return-value	Returns true if the Telewindows session supports the native user interface.

Description

Causes the specified Telewindows window to use the native user interface.

21

Operator Messages

Describes general user interface procedures to generate operator messages.

Introduction

If GEVM is loaded, these APIs create and delete gevm-message instances; otherwise, they post messages to the built-in common Message Browser of a given client. Thus, you can use these APIs to remove the dependency on GEVM so you can use a different messaging system.

For details on GEVM messages, see Chapter 4 “Event Types and Operations” in the *G2 Event Manager User’s Guide*.

grtl-delete-action-done

Synopsis

grtl-delete-action-done
(*category*: text, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes an action-done message.

grtl-delete-advisory

Synopsis

grtl-delete-advisory
(*category*: text, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes an advisory message.

grtl-delete-alarm

Synopsis

grtl-delete-alarm
(*category*: text, *target-item*: item, *client:* ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes an alarm message.

grtl-delete-calculated-alarm

Synopsis

grtl-delete-calculated-alarm
(*category*: text, *target-item*: item, *client:* ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a calculated alarm message.

grtl-delete-change-in-process-state-alarm

Synopsis

grtl-delete-change-in-process-state-alarm
(*category*: text, *target-item*: item, *client:* ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a change-in-process-state alarm message.

grtl-delete-conclusion

Synopsis

grtl-delete-conclusion
(*category*: text, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a conclusion message.

grtl-delete-deviation-alarm

Synopsis

grtl-delete-deviation-alarm
(*category*: text, *target-item*: item, *client:* ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a deviation alarm message.

grtl-delete-discrete-signal-alarm

Synopsis

grtl-delete-discrete-signal-alarm
(*category*: text, *target-item*: item, *client:* ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a discrete signal alarm message.

grtl-delete-error

Synopsis

grtl-delete-error
    (*category*: text, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes an error message.

grtl-delete-external-fault

Synopsis

grtl-delete-external-fault
(*category*: text, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes an external fault message.

grtl-delete-inferred-fault

Synopsis

grtl-delete-inferred-fault
    (*category*: text, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes an inferred fault message.

grtl-delete-instrumentation-alarm

Synopsis

grtl-delete-instrumentation-alarm
(*category*: text, *target-item*: item, *client:* ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a instrumentation alarm message.

grtl-delete-internal-fault

Synopsis

grtl-delete-internal-fault
(*category*: text, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes an internal fault message.

grtl-delete-limit-alarm

Synopsis

grtl-delete-limit-alarm
(*category*: text, *target-item*: item, *client:* ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a limit alarm message.

grtl-delete-message

Synopsis

grtl-delete-message
    (*category*: text, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a message.

grtl-delete-rate-of-change-alarm

Synopsis

grtl-delete-rate-of-change-alarm
(*category*: text, *target-item*: item, *client:* ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a rate-of-change alarm message.

grtl-delete-root-cause

Synopsis

grtl-delete-root-cause
    (*category*: text, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a root cause message.

grtl-delete-todo-action

Synopsis

grtl-delete-todo-action
(*category*: text, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a todo action message.

grtl-delete-warning

Synopsis

grtl-delete-warning
    (*category*: text, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
target-item	The target item for the message.
client	The client, such as a G2 window.

Description

Deletes a warning message.

grtl-post-action-done

Synopsis

grtl-post-action-done
(*category*: text, *message-key*: symbol, *msg-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts an action-done message. See grtl-post-message for a description.

grtl-post-advisory

Synopsis

grtl-post-advisory
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts an advisory message. See grtl-post-message for a description.

grtl-post-alarm

Synopsis

grtl-post-alarm
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts an alarm message. See grtl-post-message for a description.

grtl-post-calculated-alarm

Synopsis

grtl-post-calculated-alarm
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts a calculated alarm message. See grtl-post-message for a description.

grtl-post-change-in-process-state-alarm

Synopsis

grtl-post-change-in-process-state-alarm
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts a change-in-process-state alarm message. See grtl-post-message for a description.

grtl-post-conclusion

Synopsis

grtl-post-conclusion
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts a conclusion message. See grtl-post-message for a description.

grtl-post-deviation-alarm

Synopsis

grtl-post-alarm
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts a deviation alarm message. See grtl-post-message for a description.

grtl-post-discrete-signal-alarm

Synopsis

grtl-post-discrete-signal-alarm
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts a discrete alarm message. See grtl-post-message for a description.

grtl-post-error

Synopsis

grtl-post-error
(*message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts an error message. See grtl-post-message for a description.

grtl-post-external-fault

Synopsis

grtl-post-external-fault
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts an external fault message. See grtl-post-message for a description.

grtl-post-inferred-fault

Synopsis

grtl-post-inferred-fault
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts an inferred fault message. See grtl-post-message for a description.

grtl-post-instrumentation-alarm

Synopsis

grtl-post-instrumentation-alarm
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts a instrumentation alarm message. See grtl-post-message for a description.

grtl-post-internal-fault

Synopsis

grtl-post-internal-fault
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts an internal fault message. See grtl-post-message for a description.

grtl-post-limit-alarm

Synopsis

grtl-post-limit-alarm
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts a limit alarm message. See grtl-post-message for a description.

grtl-post-log

Synopsis

grtl-post-log
(*message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts a log message. See grtl-post-message for a description.

grtl-post-message

Synopsis

grtl-post-message
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Sends a message. If the GEVM module is loaded, this procedure creates a gevm-message. Otherwise, it posts the message to the common Message Browser of the specified client.

You can reroute these messages to your custom implementation by registering your custom callback in the grtl-module-settings object. To do this, you would create a custom callback procedure similar to the following, which gets called when a message is created or updated:

```
my-post-message
    (type: symbol, category: text, message-key: symbol,
    message-args: sequence, detail-key: symbol, detail-args: sequence,
    advice-key: symbol, advice-args: sequence, initiating-item: class item,
    target-item: class item, priority: integer, life-time: integer,
    client: class ui-client-item)
begin
end
```

Below is an empty procedure called when a message is deleted:

```
my-delete-message
    (type: symbol, category: text, target-item: class item,
    client: class ui-client-item)
begin
end
```

The code below registers the custom procedures:

```
grtl-settings: class grtl-module-settings;
begin grtl-settings = call gfr-get-active-setting(the symbol grtl-module-settings,
    client);
conclude that the post-operator-message-procedure of grtl-settings =
    the symbol my-post-message;
conclude that the delete-operator-message-procedure of grtl-settings =
    the symbol my-delete-message;
```

grtl-post-message-notification

Synopsis

grtl-post-message-notification
(*initiating-item*: item, *target-item*: item, *message*: text, *client*: ui-client-item)

Argument	Description
message	A text of the message.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts a notification message. See grtl-post-message for a description.

grtl-post-notification

Synopsis

grtl-post-notification
(*message*: text, *user-mode*: symbol)

Argument	Description
message	The text to post as a message
user-mode	Use all to post for all users or specify a user mode to restrict posting to specific users only.

Description

Posts a notification message. See grtl-post-message for a description.

grtl-post-rate-of-change-alarm

Synopsis

grtl-post-rate-of-change-alarm
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	The message category.
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts a rate-of-change alarm message. See grtl-post-message for a description.

grtl-post-notification-to-common-message-board

Synopsis

grtl-post-notification-to-common-message-board
(*message*: text, *user-mode*: symbol)

Argument	Description
message	The message to post.
user-mode	Only post the message if he is in this user mode.

Description

Posts a message to the common Message Board and displays it on the screen.

grtl-post-root-cause

Synopsis

grtl-post-root-cause
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	
message-key	
msg-args	
detail-key	
detail-args	
advice-key	
advice-args	
initiating-item	
target-item	
client	

Description

Posts a root cause message. See grtl-post-message for a description.

grtl-post-todo-action

Synopsis

grtl-post-todo-action
(*category*: text, *message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
category	
message-key	
msg-args	
detail-key	
detail-args	
advice-key	
advice-args	
initiating-item	
target-item	
client	

Description

Posts an action-todo message. See grtl-post-message for a description.

grtl-post-warning

Synopsis

grtl-post-warning
(*message-key*: symbol, *message-args*: sequence, *detail-key*: symbol, *detail-args*: sequence, *advice-key*: symbol, *advice-args*: sequence, *initiating-item*: item, *target-item*: item, *client*: ui-client-item)

Argument	Description
message-key	A lookup key for the message.
message-args	A sequence of simple values that will be inserted into the message text.
detail-key	A lookup key for the message detail.
detail-args	A sequence of simple values that will be inserted into the message text.
advice-key	A lookup key for the message advice.
advice-args	A sequence of simple values that will be inserted into the advice text.
initiating-item	The item sending the message, which must be a type of gevm-event-initiating-item.
target-item	The target item for the message.
client	The client of this message invocation, such as a G2 window.

Description

Posts a warning message. See grtl-post-message for a description.

22

Menus and Toolbars

Describes procedures to manage menus, toolbars, status bars, and keyboard shortcuts.

Introduction

This chapter describes classes, procedures, and methods related to building menu bars, popup menus, toolbars, status bars, and keyboard shortcuts. If the Telewindows session supports the native user interface, the APIs use the Native

Menu System (NMS) APIs; otherwise, they use the GMS APIs. Toolbars are only supported in the native user interface.

This implementation minimizes the overhead required by building menus by using the higher-level object-oriented API, such as the G2 Menu System (GMS). It also supports other important user interface functionality, such as mnemonics in menus. Finally, it provides dynamic configuration of menus and toolbars by using an object-oriented approach, which makes it easy to build menus dynamically, based on the application context, for example, the selected view, the selected objects, the user mode, and so on.

The default menu bar and toolbars included in GRTL are accessible from the Programmer's Interface button on the grtl-top-level workspace.

The *g2i-ui-demo.kb* provides simple examples of how to customize or extend the GRTL user interface.

Configuring Menus and Toolbars on Startup

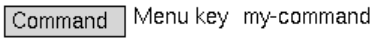
You can configure the attributes of menu bars, toolbars, and status bars in the configuration file, *config.txt*, by default.

To configure menu bars, toolbars, and status bars, the section name in the configuration file must match the name of the menu bar, toolbar, or status bar, respectively. Under each section, configure the following attributes of the menu bar, toolbar, or status bar, respectively:

- Menu bars — Configure any attribute of the menu bar.
- Toolbars — Configure the following attributes of a toolbar:
  - enabled
  - initially-visible
  - initial-dock
  - initial-dock-priority
  - neighbor-dock
  - neighbor-toolbar
  - enable-tooltips
- Status bars — Configure the following attributes of a status bar:
  - enabled
  - initially-visible
  - minimum-height

Menu Commands

You create a menu command by creating an instance of grtl-menu-command-procedure, which you can clone from the User Interface palette of the G2 toolbox:



Menu commands are uniquely identified by a symbolic name. This symbolic name is supplied when creating menu entries in the menu bar, popup menu, toolbar, or shortcut bar, which activates the same execution regardless of how the command has been activated and the underlying technology, native user interface or GMS, for example.

To execute commands, GRTL calls the procedure that is named in the menu-command-execute-procedure attribute of the grtl-user-preferences associated with the G2 or Telewindows sessions. The default procedure processes standard menu commands and dispatches others as follows:

1. If an instance of a grtl-menu-command-procedure (a subclass of a procedure), such that the menu-key attribute of that instance corresponds to the symbolic name of the command, it calls the grtl-menu-command-procedure instance using the following signature:

   ```
   my-command
   (menu-key: symbol, selected-wks: item-or-value,
   selected-items: sequence, selected-items-excluding-connections: sequence,
   menu-info: item-or-value, options: structure, win: class ui-client-item)
   ```

   Instances of grtl-menu-command-procedure are considered global commands that typically do not apply to the currently selected item. Examples are Load, Save, View.

2. If no grtl-menu-command-procedure is found, the command is dispatched to the currently selected item by calling the _grtl-item-exec-menu-callback method. You may create methods for your specific classes and commands to implement the desired behavior. The signature of this callback is:

   ```
   my-item-exec-menu-callback
   (target: class my-class, menu-cmd: symbol, menu-handle: item-or-value,
   selected-wks: item-or-value, options: item-or-value,
   win: class ui-client-item)
   ```

Menu Entries

GRTL provides APIs for:

- Adding Menu Entries

- Adding Menu Entries with a Resource Group
- Updating Menu Entries

Adding Menu Entries

The following APIs create menu entries for the menu bar, popup menus, and toolbars.

Each of these APIs accesses the GRTL localization resource to collect information to configure the menu item. They look for the following information, where *menu-key* is the menu name:

- To get the menu label:

  MENU.*menu-key*.LABEL, "My Menu"

- To get the mnemonic letter within the label:

  MENU.*menu-key*.MNEMONIC, "M"

- To get the keyboard accelerator associated with the menu:

  MENU.*menu-key*.ACCELERATOR, "F9"

- To get the tooltip associated with the menu:

  MENU.*menu-key*.TOOLTIP, "My Custom Menu"

grtl-add-menu-item
(*menu-key*: symbol, *custom-label*: text, *show-without-label*: truth-value, *menu-icon*: symbol, *menu-enabled*: truth-value, *menu-checked*: truth-value, *parent-menu*: item-or-value, *menu-commands*: class hash-table, *win*: class ui-client-item)
-> *handle*: integer

Adds a menu entry.

grtl-add-static-cascading-menu-item
(*menu-key*: symbol, *custom-label*: text, *show-without-label*: truth-value, *menu-icon*: symbol, *sub-menu-constructor-procedure*: symbol, *parent-menu*: item-or-value, *menu-commands*: class hash-table, *win*: class ui-client-item)
-> *handle*: integer

Adds a static cascading menu entry. The submenus are defined at menu creation time and are not updated each time the cascading menu is displayed. Note that for GMS, the menus are always dynamic "under the hood."

grtl-add-dynamic-cascading-menu-item
(*menu-key*: symbol, *custom-label*: text, *show-without-label*: truth-value, *menu-icon*: symbol, *sub-menu-constructor-procedure*: symbol, *parent-menu*: item-or-value, *menu-commands*: class hash-table,

win: class ui-client-item)
-> *handle*: integer

Adds a dynamic cascading menu.

grtl-add-built-in-menu-item
(*menu-key*: symbol, *custom-label*: text, *show-without-label*: truth-value,
menu-icon: symbol, *menu-enabled*: truth-value, *menu-checked*: truth-value,
parent-menu: item-or-value, *menu-commands*: class hash-table,
win: class ui-client-item)
-> *handle*: integer

Adds a G2 build in menu. This procedure only works for the Windows Native Menu System (NMS). The *menu-key* is a symbol of the form:

MENU.*menu-key*.LABEL | MNEMONIC | ACCELERATOR | TOOLTIP

You can reuse the *menu-key* symbols in the `resources-english.txt` file in the GRTL APIs to add menu items. To search for the relevant menu key in the `resources-english.txt` file, we recommend using the menu label. For example, if you want the *menu-key* corresponding to the popup menu option labeled "Add stubs", search for "Add stubs" in the text file.

To add menu options using a custom resource text file, that is, something other than `resources-english.txt`, see grtl-add-menu-item-using-resource-group.

grtl-add-combo-box-menu-item
(*menu-key*: symbol, *custom-label*: text, *show-without-label*: truth-value,
choices: sequence, *initial-value*: text, *width*: integer,
parent-menu: item-or-value, *menu-commands*: class hash-table,
win: class ui-client-item)
-> *handle*: integer

Adds a combo box to a menu or toolbar. Combo boxes are typically used in toolbars.

grtl-add-edit-box-menu-item
(*menu-key*: symbol, *current-value*: text, *width*: integer,
parent-menu: item-or-value, *menu-commands*: class hash-table,
win: class ui-client-item)
-> *handle*: integer

Adds an edit box to a menu or toolbar. Edit boxes are typically used in toolbars.

grtl-add-separator-menu-item
(*parent-menu*: item-or-value, *menu-commands*: class hash-table,
win: class ui-client-item)
-> *handle*: integer

Adds a separator to the menu.

Adding Menu Entries with a Resource Group

The following APIs create menu entries for the menu bar, popup menus, and toolbars, using a specific resource group.

grtl-add-menu-item-using-resource-group
(*menu-key*: symbol, *custom-label*: text, *show-without-label*: truth-value, *menu-icon*: symbol, *menu-enabled*: truth-value, *menu-checked*: truth-value, *resource-group-name*: symbol, *parent-menu*: item-or-value, *menu-commands*: class hash-table, *win*: class ui-client-item)
-> *handle*: integer

Adds a menu entry, using a GFR resource group.

grtl-add-static-cascading-menu-item-using-resource-group
(*menu-key*: symbol, *custom-label*: text, *show-without-label*: truth-value, *menu-icon*: symbol, *sub-menu-constructor-procedure*: symbol, *resource-group-name*: symbol, *parent-menu*: item-or-value, *menu-commands*: class hash-table, *win*: class ui-client-item)
-> *handle*: integer

Adds a static cascading menu entry, using a GFR resource group. The submenus are defined at menu creation time and are not updated each time the cascading menu is displayed. Note that for GMS, the menus are always dynamic "under the hood."

grtl-add-dynamic-cascading-menu-item-using-resource-group
(*menu-key*: symbol, *custom-label*: text, *show-without-label*: truth-value, *menu-icon*: symbol, *sub-menu-constructor-procedure*: symbol, *resource-group-name*: symbol, *parent-menu*: item-or-value, *menu-commands*: class hash-table, *win*: class ui-client-item)
-> *handle*: integer

Adds a dynamic cascading menu, using a GFR resource group.

grtl-add-built-in-menu-item-using-resource-group
(*menu-key*: symbol, *custom-label*: text, *show-without-label*: truth-value, *menu-icon*: symbol, *menu-enabled*: truth-value, *menu-checked*: truth-value, *resource-group-name*: symbol, *parent-menu*: item-or-value, *menu-commands*: class hash-table, *win*: class ui-client-item)
-> *handle*: integer

Adds a G2 build in menu, using a GFR resource group. This procedure only works for the Windows Native Menu System (NMS).

grtl-add-combo-box-menu-item-using-resource-group
(*menu-key*: symbol, *custom-label*: text, show-without-label: truth-value, *choices*: sequence, *initial-value*: text, *width*: integer, *resource-group-name*: symbol, *parent-menu*: item-or-value, *menu-commands*: class hash-table, *win*: class ui-client-item)
-> *handle*: integer

Adds a combo box to a menu or toolbar, using a GFR resource group. Combo boxes are typically used in toolbars.

grtl-add-edit-box-menu-item-using-resource-group
(*menu-key*: symbol, *current-value*: text, *width*: integer,
resource-group-name: symbol, *parent-menu*: item-or-value,
menu-commands: class hash-table, *win*: class ui-client-item)
-> *handle*: integer

Adds an edit box to a menu or toolbar, using a GFR resource group. Combo boxes are typically used in toolbars.

Updating Menu Entries

The following APIs are available to change the state of menu entries:

grtl-update-menu-item
(*menu-key*: symbol, *change-enabled*: truth-value, *is-enabled*: truth-value,
change-checked: truth-value, *is-checked*: truth-value, *win*: class ui-client-item)

Updates the specified menu by enabling/disabling and/or checking/unchecking the menu for the specified window. This API updates every menu-item using the *menu-key* in the menu bar, popup menus, and toolbars with a single call.

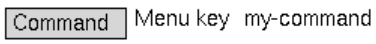
grtl-update-edit-box-or-combo-box-menu
(*menu-key*: symbol, *new-text*: text, *win*: class ui-client-item)

Updates every edit box and combo box menu entry identified by *menu-key* in the toolbar, popup menu, s and toolbars for the specified window.

Menu Bars

The name of the menu bar to display for each user is specified in the menubar-name attribute of the user preference. This symbolic name can either refer to the gms-user-key of a gms-menu-bar-template or the menubar-name of a grtl-menubar instance.

You create a menu bar by creating an instance of grtl-menubar, which you can clone from the User Interface palette of the G2 toolbox:



The menubar-name attribute of the grtl-menubar instance specifies the name of the menubar and should be unique. The populate-procedure of the grtl-menubar refers to the procedure to use for building the content by adding menu items by using the APIs described earlier.

grtl-menubar

Class Inheritance Path

grtl-menubar, object

Attributes

Attribute	Description
menubar-name	The name of the menu bar, which must be unique.
Allowable values:	symbol
Default value:	g2
enabled	Whether the menu bar is enabled.
Allowable values:	truth-value
Default value:	true
populate-procedure	The name of a G2 procedure that is called to populate the menu bar. The populate procedure should use the APIs described in Menu Entries. The signature of the procedure is: populate-menubar (*menubar*: integer, *user*: class grtl-user-preferences, *menu-commands*: class hash-table, *win*: class ui-client-item)
Allowable values:	symbol
Default value:	unspecified

update-procedure

The name of a G2 procedure that is called to update the state of menu items in the menubar, popup menus, and toolbars. This procedure is called typically upon selection change. To update menu entries the specified update procedure should use the APIs grtl-update-menu-item or grtl-update-edit-box-or-combo-box-menu.

The signature of the procedure is:

update-menubar-and-toolbars
(*user*: class grtl-user-preferences,
session: class grtl-user-session,
selected-worspace: item-or-value,
selected-items: sequence,
count-of-selected-items: integer,
selected-items-without-connections: sequence,
count-of-selected-items-without-connections: integer,
target: item-or-value,
canResize: truth-value,
canClone: truth-value,
canTransfer: truth-value,
canDelete: truth-value,
canChangeLayout: truth-value,
isPrivate: truth-value,
menu-commands: class hash-table,
win: class ui-client-item)

Allowable values: symbol

Default value: unspecified

execute-command-procedure

The name of a G2 procedure that is called to execute a menu command specified by its *menu-key*. Custom procedures should call the default procedure called _grtl-menu-callback to ensure correct processing of default built-in menu commands. Also note that the default procedure _grtl-menu-callback calls the following method to request the execution of many commands:

```
_grtl-item-exec-menu-callback
(target: class item, menu-cmd: symbol,
menu-handle: item-or-value,
selected-workspace: item-or-value,
options: item-or-value,
win: class ui-client-item)
```

Therefore, typically instead of specifying a custom procedure in this attribute, it is better to implement the method _grtl-item-exec-menu-callback for custom classes.

The signature of the procedure is:

```
execute-command-procedure
(menu-key: symbol,
default-target: item-or-value,
menu-info: item-or-value, options: structure,
win: class ui-client-item)
```

Allowable values: symbol

Default value: _grtl-menu-callback

visible-in-user-modes

A sequence of symbols that specify the user modes in which the menu bar is visible.

Allowable values: sequence

Default value: sequence()

Note that for the GMS implementation, the menu bar is specified by using the standard GMS graphical language, but typically only the menu bar items are specified and not submenus of each entry in the menu bar. The gms-user-key of the gms-menu-bar-template specifies the menu bar name, using the same name used in the menubar-name of the grtl-menubar instance. Entries in the menu bar are specified using grtl-dynamic-cascade-template instances. The callback-procedure-name of these instances can refer to the same procedure used to

populate the cascading menus for the native menu bar specified in the populate-procedure of the grtl-toolbar instance.

Please refer to the GRTL default menu bar implementation for examples. It includes both a GMS menu bar and a grtl-menubar instance.

Below is a list of APIs, but typically developers do not need to call them because the GRTL user interface environment, if enabled, calls them to configure the user interface upon connection:

grtl-create-menubar
(*user*: class grtl-user-preferences, *session*: class grtl-user-session, *win*: class ui-client-item)

Creates the menubar specified in the user's user preference. Displays the menubar in the user modes specified in visible-in-user-modes.

grtl-delete-menubar
(*user*: class grtl-user-preferences, *session*: class grtl-user-session, *win*: item-or-value)

Deletes the menubar.

grtl-show-or-hide-menubar
(*user*: class grtl-user-preferences, *session*: class grtl-user-session, *show-menubar*: truth-value, *win*: class ui-client-item)

Shows or hides the menubar.

grtl-update-menus-and-toolbars
(*win*: class ui-client-item, *reason*: symbol)

grtl-update-menus-and-toolbars
(*user*: class grtl-user-preferences, *session*: class grtl-user-session, *reason*: symbol, *win*: class ui-client-item)

Updates the state of menus and toolbars such as enabling/disabling them or checking/unchecking them. These two methods call the update procedure of menubar and toolbars. The *reason* argument can be the symbol menu-execution, user-mode-changed, selection-changed, hide, show, or refresh. This argument can be used to narrow down the specific actions to update the menubar and toolbars. The refresh option forces an immediate update of the menus and toolbars; otherwise, updates are collapsed in time buckets of 50 ms to avoid overhead. Note that these methods are automatically called by the GRTL user interface framework as the selection changes.

Popup Menus

Popup menus are typically displayed when the user right clicks an items and are typically specific for each class, though some common menu entries exist for all classes. GRTL uses an object-oriented approach by providing a method for each class to populate the content of the popup menu. Typically, this method first calls the method for the superior class to start with its menu choices. Similar to the menu bar, this method uses the APIs described in Menu Entries to populate the popup menu.

Upon selection of a menu item in a popup menu, the same GRTL execution path is used as if the command were selected in the menu bar by calling the execute-command-procedure specified in the menubar. See grtl-menubar for details.

The user menu choice Popup Menu is triggered when right clicking on an item in all user modes except administrator. This menu choice calls the following method on the target item. If multiple items are selected, it is handled internally by GRTL and displays an appropriate menu for this situation. You can create methods for your subclasses to implement your own additional menus. Calling next method ensures that the default menus from the parent class are inherited. Your menus are added after the predefined menus.

_grtl-create-popup-menu
(*target*: class item, *display-target*: symbol,
cascading-property-menu: truth-value, *user*: class grtl-user-preferences,
parent-menu: item-or-value, *menu-commands*: class hash-table,
win: class ui-client-item)

The *display-target* is the symbol item if displayed on an item or the symbol tree-view if displayed in a tree view.

Here is an example:

```
_grtl-create-popup-menu (target: class my-class, display-target: symbol,
    cascading-property-menu: truth-value, user: class grtl-user-preferences,
    parent-menu: item-or-value, menu-commands: class hash-table,
    win: class g2-window)
handle: integer;
isPrivate: truth-value;
begin

    call next method;

    isPrivate = call grtl-is-item-private(target);

    { --- Implement the properties popup menu choice }
    handle = call grtl-add-menu-item (the symbol grtl-make-permanent, "", false,
        the symbol none, not(isPrivate) and target exists, false, parent-menu,
        menu-commands, win);
 end
```

Below is a list of APIs, but typically developers do not need to call them because the GRTL user interface environment if enabled calls them to configure the user interface upon connection:

grtl-display-popup-menu
(*target*: class item, *win*: class ui-client-item)

Creates a popup menu and displays the popup menu. This procedure calls grtl-create-and-display-popup-menu() after determining the best location of the popup menu in the screen.

grtl-create-and-display-popup-menu
(*target*: class item, *display-target*: symbo, *x-pos*: integer, *y-pos*: integer, *win*: class ui-client-item)

Creates a popup menu by calling _grtl-create-popup-menu for the target item (see above for details) and displays the popup menu. If the *display-target* is tree-view, this procedure adds the go-to menu choice at the beginning of the popup menu. This procedure also adds the label of G2 user-menu-choice instances assigned to the target item at the end of the popup menu list. The user-menu-choices are sorted in alphabetical order.

Toolbar

You create a toolbar by creating an instance of grtl-toolbar, which you can clone from the User Interface palette of the G2 toolbox:

The toolbar-name attribute specifies the name of the toolbar and should be unique. The populate-procedure of the grtl-toolbar instance refers to the procedure that builds the content, which uses the same APIs than the ones used to create menu items for menu bars or toolbars.

Upon selection of a menu item in a toolbar, the same GRTL execution path is used as if the command were selected in the menu bar by calling the execute-command-procedure specified in the menubar. See grtl-menubar for details.

To provide localization for toolbars, as well as other information, GRTL collects additional information regarding the toolbar from the GRTL localization resource. It looks for the following information, where *toolbar-name* is the name assigned to the toolbar:

- To get the label of the toolbar:

  TOOLBAR.*toolbar-name*.LABEL, "Fault Modeling"

- To get the mnemonic letter within the toolbar label:

  TOOLBAR.*toolbar-name*.MNEMONIC, "F"

grtl-toolbar

Class Inheritance Path

grtl-toolbar, object

Attributes

Attribute	Description
toolbar-name	The name of the toolbar, which must be unique.
Allowable values:	symbol
Default value:	g2
enabled	Whether the toolbar is enabled.
Allowable values:	truth-value
Default value:	true
initially-visible	If true, displays the toolbar upon connection.
Allowable values:	truth-value
Default value:	true
visible-user-modes	A sequence of symbols listing the user modes specifying when the toolbar should be visible.
Allowable values:	sequence of symbols
Default value:	sequence( )
initial-dock	The initial dock position of the toolbar, as a symbol.
Allowable values:	top, bottom, left, right, or float
Default value:	top
initial-dock-priority	Upon connection all visible toolbars are sorted by their initial-dock-priority and created in this order. Toolbars with a lower initial-dock-priority are created first.

Allowable values: integer

Default value: 0

neighbor-dock The dock style relative to the neighbor toolbar, if visible.

Allowable values: top, bottom, left, or right

Default value: left

neighbor-toolbar The toolbar-name of the to the neighbor toolbar.

Allowable values: symbol

Default value: none

enable-tooltips If true, enables tooltips for the toolbar.

Allowable values: truth-value

Default value: true

button-style The button style of the menu entries in the toolbar, as a symbol.

Allowable values: automatic, icon-only, caption-only, icon-and-caption, or default

Default value: automatic

populate-procedure The name of a G2 procedure that is called to populate the toolbar. The populate procedure should use the APIs described in Menu Entries.

The signature of the procedure is:

```
populate-toolbar
  (toolbar: integer,
  user: class grtl-user-preferences,
  menu-commands: class hash-table,
  win: class ui-client-item)
```

Allowable values: symbol

Default value: unspecified

update-procedure

The name of a G2 procedure that is called to update the state of menu items in the toolbar. This procedure is called typically upon selection change but is typically not required if the menus are included in the menubar since the update procedure of the menu bar updates menus with the same menu-key in toolbars as well. To update menu entries the specified update procedure should use the APIs grtl-update-menu-item or grtl-update-edit-box-or-combo-box-menu.

The signature of the procedure is:

```
update-toolbar
  (user: class grtl-user-preferences,
  session: class grtl-user-session,
  selected-workspace: item-or-value,
  selected-items: sequence,
  count-of-selected-items: integer,
  selected-items-without-connections:
sequence,
count-of-selected-items-without-connections:
integer,
  target: item-or-value,
  canResize: truth-value,
  canClone: truth-value,
  canTransfer: truth-value,
  canDelete: truth-value,
  canChangeLayout: truth-value,
  isPrivate: truth-value,
  menu-commands: class hash-table,
  win: class ui-client-item)
```

Allowable values: symbol

Default value: unspecified

GRTL includes three predefined toolbars: the standard toolbar, the layout toolbar, and the Web toolbar. Refer to these predefined toolbars for examples on how to configure them. They are accessible from the subworkspace of the Programmer's Interface of the grtl-top-level workspace.

Below is a list of APIs, but typically developers do not need to call them because the GRTL user interface environment, if enabled, calls them to configure the user interface upon connection:

grtl-create-toolbars
(*user*: class grtl-user-preferences, *session*: class grtl-user-session, *win*: class ui-client-item)

Creates toolbar instances for all grtl-toolbar instances with initially-visible set to true.

grtl-delete-toolbars
(*user*: class grtl-user-preferences, *session*: class grtl-user-session, *win*: item-or-value)

Deletes toolbar instances associated with the user's session

Status Bars

You create a status bar by creating an instance of grtl-status-bar, which you can clone from the User Interface palette of the G2 toolbox.

grtl-status-bar

Class Inheritance Path

grtl-status-bar, object

Attributes

Attribute	Description
status-bar-name	The name of the status bar.
Allowable values:	symbol
Default value:	g2, with an index
enabled	Whether the status bar is enabled.
Allowable values:	truth-value
Default value:	true
initially-visible	When true, the status bar is initially visible.
Allowable values:	truth-value
Default value:	true
visible-in-user-modes	The list of user modes in which the status bar is visible.
Allowable values:	sequence
Default value:	sequence()
minimum-height	The minimum height for the status bar.
Allowable values:	integer
Default value:	-1
callback-procedure	A procedure to call when the user clicks the status bar.
Allowable values:	symbol
Default value:	unspecified

panes

The list of panes to show in the status bar, as a sequence of structures, where each structure has this syntax:

structure
(id: *symbol*
 text: *text*,
 width: *symbol*,
 alignment: *symbol*
 visible: *truth-value*,
 enabled: *truth-value*,
 borders: *truth-value*)

Allowable values: sequence

Default value:
```
sequence
  (structure (
     id: the symbol network-info,
     text: "localhost:1111",
     width: the symbol fit,
     alignment: the symbol center,
     visible: true,
     enabled: true,
     borders: true),
   structure (
     id: the symbol gevm-message,
     text: "",
     width: 250,
     alignment: the symbol left,
     visible: true,
     enabled: true,
     borders: true))
```

Keyboard Shortcuts

You can configure keyboard shortcuts for menus, which are read from a text resource localization file. The keyboard accelerators are defined separately from their associated menus.

You create a keyboard shortcut by creating an instance of grtl-keyboard-accelerator, which you can clone from the User Interface palette of the G2 toolbox:

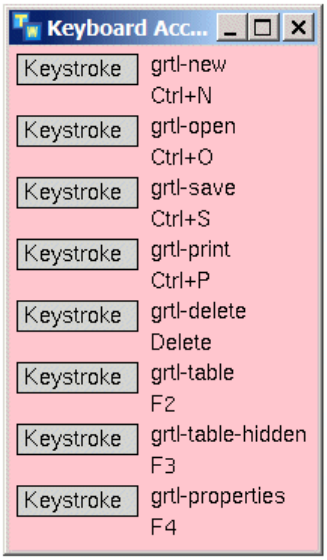
The keystroke is extracted from the GRTL resource localization file, just like it is for menus.

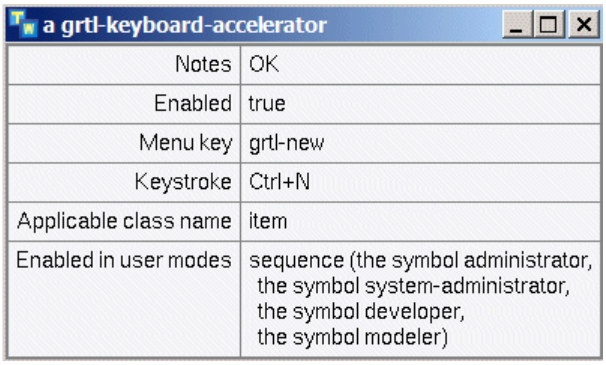
GRTL provides a generic user menu choice called _keystroke-handler to process keyboard accelerators. This user menu choice calls g2-last-input-event to determine the keystroke and process it. Note that similar to popup menus, keystrokes are executed on the selected item(s) or workspace, not on the item under the mouse, which is the typical Microsoft/Java behavior.

To use the keystroke handler, use item configuration statements such as this:

```
configure the user interface as follows:
when in system-administrator, developer, or modeler mode:
typing "control+n" on any item implies _keystroke-handler;
```

For example, here are the built-in keyboard accelerators for GRTL and a sample attribute table:





Notes	OK
Enabled	true
Menu key	grtl-new
Keystroke	Ctrl+N
Applicable class name	item
Enabled in user modes	sequence (the symbol administrator, the symbol system-administrator, the symbol developer, the symbol modeler)

grtl-keyboard-accelerator

Class Inheritance Path

grtl-keyboard-accelerator, object

Attributes

Attribute	Description
enabled	Specifies whether the accelerator is enabled
Allowable values:	truth-value
Default value:	true
enabled-in-user-modes	A sequence of user modes, as symbols, or an empty sequence, to determine the user modes in which the keystroke is enabled.
Allowable values:	sequence
Default value:	sequence()
menu-key	The symbolic action/command name used at execution time. The menu-key is typically the menu name to ensure it provides the accelerator for the correct menu. When the accelerator is enabled, the menu-key is used to configure the keyboard accelerator based on the content of the GRTL text localization resource.
Allowable values:	symbol
Default value:	none
applicable-class-name	Constraints the menu choice to some classes.
Allowable values:	symbol
Default value:	none

keystroke Automatically filled in with the value extracted from the GRTL localization file.

Allowable values: symbol

Default value: none

23

Project Menu and Navigator

Describes a class to create and organize menu choices in the Project menu, as well as the structure of the Navigator and the behavior of the Manage dialog.

Introduction

Most applications include a navigation tree view to organize the content of applications. GRTL provides a flexible way of specifying the contents of the Project menu and the Navigator, and the associated Manage dialog in the Project menu, without requiring you to write code. This feature allows you to easily and dynamically extend the contents as modules are merged in or to insert application-specific nodes in the hierarchy. GRTL provides a pre-build Navigator that is shared and extended across applications products and demos.

Project Menu and Navigator Classes

To support this feature, GRTL provides the following classes to describe the contents of the Project menu and the Navigator:

- grtl-node-specification
- grtl-generic-node-specification
- grtl-instance-hierarchy-node-specification
- grtl-class-hierarchy-node-specification
- grtl-module-hierarchy-node-specification
- grtl-workspace-hierarchy-node-specification
- grtl-separator-node-specification

Application Navigator Organization

Note all nodes might not be visible since the modules might not be merged in. The layout is based on typical G2 application patterns: the managed system, the managed system model, and the management logic. The managed system is the real system being stabilized, controlled, or optimized. The management logic is the reasoning that will be used to achieve these objectives. A managed system model is a model of the managed system. As in every real world application, G2 plays the role of a managed system model or the management logic or both. The sections below explain these concepts so that you can use the demonstration applications to communicate most effectively the use of G2 in real-world situations.

Managed System

The managed system is the real system being stabilized, controlled, or optimized. In a process control application, the managed system is a reactor or group of production processes or a whole plant. In a supply chain application, the managed system is the supply chain. In a manufacturing execution system application, the managed system is the scope of production processes to be synchronized. In a telecom application, the managed system is the network of machines or equipment or software applications. And in a business rules application, the managed system is a business process for supply, production, or delivery of a good or service.

Managed System Model

G2 is often used to develop a model of the managed system. All of the demo applications include a managed system model to reinforce the importance of the

model, both for exploring scenarios and in real-time management. Examples of managed system models are supply chain models, for example, e-SCOR models, production process models, for example, NOL software sensor models, business process models, for example, ReThink models, network management models, and a model of a battlespace.

Management Logic

G2 is most widely used to create the management logic to stabilize, control, or optimize a managed system. Management logic in G2 follows a sense-respond paradigm. Sense tasks have as a goal determining the state or condition of the managed system. Respond tasks have as a goal developing actions to stabilize, control, or optimize the managed system. We often subdivide sense tasks into event detection and condition diagnosis tasks. In the demo applications, you can identify sense-type logic from respond-type logic. Note that when the management logic is linked into the managed system model, it is often simulated to test or validate that the management logic will work as expected.

G2 Application Navigator Organization

For example, here is the application navigator organization for the built-in navigator in G2. Each node describes the type of children that can be created, the relevant toolboxes for the node, and the views to display for those children, which are typically modeling workspace or a properties dialog, if not other view is relevant.

System Models	**Description**
Business Processes	Generic business process models. Toolboxes: ReThink toolbox (activities, instruments, resources, etc) Views: workspace
Supply Chain Processes	Generic supply chain process models. Toolboxes: e-SCOR toolbox (roles, products, etc) Views: workspace
Manufacturing Processes	Generic manufacturing process maps. Toolboxes: Optegrity toolbox (equipment and sensors) Views: workspace

Network Diagrams	Generic computer network and devices diagrams. Toolboxes: integrity toolbox (computers, networks, devices) Views: workspace
Logic	Reasoning logic. Reasoning logic relies on diagrams or maps representing the physical model (see nodes above).
Detect	Detection logic
Dataflow Templates	Event detection dataflow templates; based on sensor information and statistical analysis on histories. Toolboxes: GEDP toolbox Views: workspace with flow diagram
Dataflow Instances	Event detection dataflow instances; based on sensor information and statistical analysis on histories. Toolboxes: GEDP toolbox Views: workspace with flow diagram
Workflows	Event detection workflows. Toolboxes: G2GL Views: workspace workflow diagram
Test	Tests triggered by the operator or started automatically by the diagnosis logic to gather more information for the purpose of diagnosis.
Dataflow Templates	Generic test dataflow logic; based on sensor information and statistical analysis on histories. Toolboxes: GEDP toolbox Views: workspace with flow diagram
Dataflow Instances	Specific test dataflow instances; based on sensor information and statistical analysis on histories. Toolboxes: GEDP toolbox Views: workspace with flow diagram

Workflows	Test workflows. Toolboxes: G2GL Views: workspace workflow diagram
Diagnose	Diagnostic logic. Toolboxes: none Views: none
Fault Models	Generic fault models. Toolboxes: SymCure toolbox Views: workspace with SymCure fault model
Diagnosis Managers	Specific fault models (generated from diagnosis manager). Toolboxes: none Views: Specific fault model
Consoles	Consoles for diagnosing specific fault models. Toolboxes: none Views: Alarms Browser, Root Causes Browser, Test Actions Browser, Repair Actions Browser
Debug	Debugging console. Toolboxes: none Views: none
Enable	Enables debugging. Toolboxes: none Views: none
Sequential	Sequential debugging mode. Toolboxes: none Views: properties
Parallel	Parallel debugging mode. Toolboxes: none Views: properties

Import	Import fault models. Toolboxes: none Views: none
Generic	Import generic fault models. Toolboxes: none Views: properties
Specific	Import specific fault models. Toolboxes: none Views: properties
Enable Tuning	Enables tuning of generic fault models. Toolboxes: none Views: none
Respond	Response actions triggered by the operator or started automatically by the diagnosis logic to auto correct problems.
Dataflow Templates	Generic response dataflow logic; based on sensor information and statistical analysis on histories. Toolboxes: GEDP toolbox Views: none
Dataflow Instances	Specific dataflow instances; based on sensor information and statistical analysis on histories. Toolboxes: GEDP toolbox Views: workspace with flow diagram
Workflows	Response workflows. Toolboxes: G2GL or ReThink Views: none
Orchestrate	BPEL/G2GL orchestration flows (business, MES like, etc). Toolboxes: G2GL toolbox Views: workspace with G2GL process (subworkspace of g2gl-process)

Business Rules	Business rules organized by rule categories. Toolboxes: none Views: none
Neural Network Models	Neural Net models.
Back Propagation Neural Net	Back Propagation Neural Net models. Toolboxes: none Views: properties
Radial Basis Function Neural Net	Radial Basis Neural Net models. Toolboxes: none Views: properties
Rho Neural Net	Rho Neural Net models. Toolboxes: none Views: properties
Autoassociative Neural Net	Auto Associative Neural Net models. Toolboxes: none Views: properties
Ensemble Neural Net	Ensemble Neural Net Models. Toolboxes: none Views: properties
Statistical Models	Statistical models.
Principal Least Square Net (PLS)	PLS models. Toolboxes: none Views: properties
Principal Component Analysis (PCA)	PCA models. Toolboxes: none Views: properties

Datapoint Replay	Datapoint replay objects; datapoints are re-played based on information specified in data series. Toolboxes: none Views: none
Datapoint Simulations	Generic node grouping for datapoint simulation; datapoint simulation is based on random generators. Toolboxes: none Views: none
Reports	Tabular Reports
Charts	Charts
Object Models	Class hierarchies of objects used to build models of the managed system.
Business Objects	Business object modeling objects. Toolboxes: none Views: properties
Business Processes	Business process modeling objects. Toolboxes: none Views: properties
Supply Chain Processes	Supply chain network modeling objects. Toolboxes: none Views: properties
Instruments and Equipment	Manufacturing equipment and sensors modeling objects. Toolboxes: none Views: properties
Networks & Devices	Computer networks and devices modeling objects. Toolboxes: none Views: properties
System Settings	System configurations in an application.
Interfaces	Adaptor interfaces organized by standard.

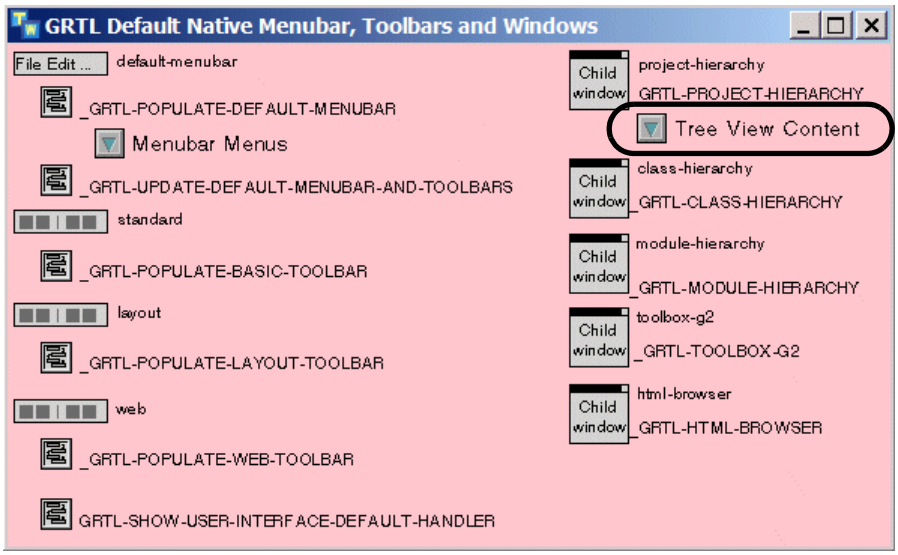
OPC	OPC adaptor interfaces. Toolboxes: none Views: properties
PI	PI adaptor interfaces. Toolboxes: none Views: properties
SQL	PI adaptor interfaces. Toolboxes: none Views: properties
SMTP	SMTP adaptor interfaces. Toolboxes: none Views: properties
JMS	JMS adaptor interfaces. Toolboxes: none Views: properties
SNMP	SMNP adaptor interfaces. Toolboxes: none Views: properties
HTTP	HTTP adaptor interfaces. Toolboxes: none Views: properties
Interface Pools	Adaptor interface pools organized by standard.
OPC	OPC adaptor interface pools. Toolboxes: none Views: properties
PI	PI adaptor interface pools. Toolboxes: none Views: properties

SQL	PI adaptor interface pools. Toolboxes: none Views: properties
SMTP	SMTP adaptor interface pools. Toolboxes: none Views: properties
JMS	JMS adaptor interface pools. Toolboxes: none Views: properties
External Datapoints	External datapoints Toolboxes: none Views: properties
Datapoint Series	Datapoint series loaded from csv files.
Continuous	Continuous datapoint series imported from a csv file. Toolboxes: none Views: properties
Differential	Differential datapoint series imported from a csv file. Toolboxes: none Views: properties
Datapoint Logs	Datapoint log information; typically stored as a data series in a csv file or a database. Toolboxes: none Views: properties
Message Browsers	Message queues and their associated browsers.
Queues	Message queues. Toolboxes: none Views: properties

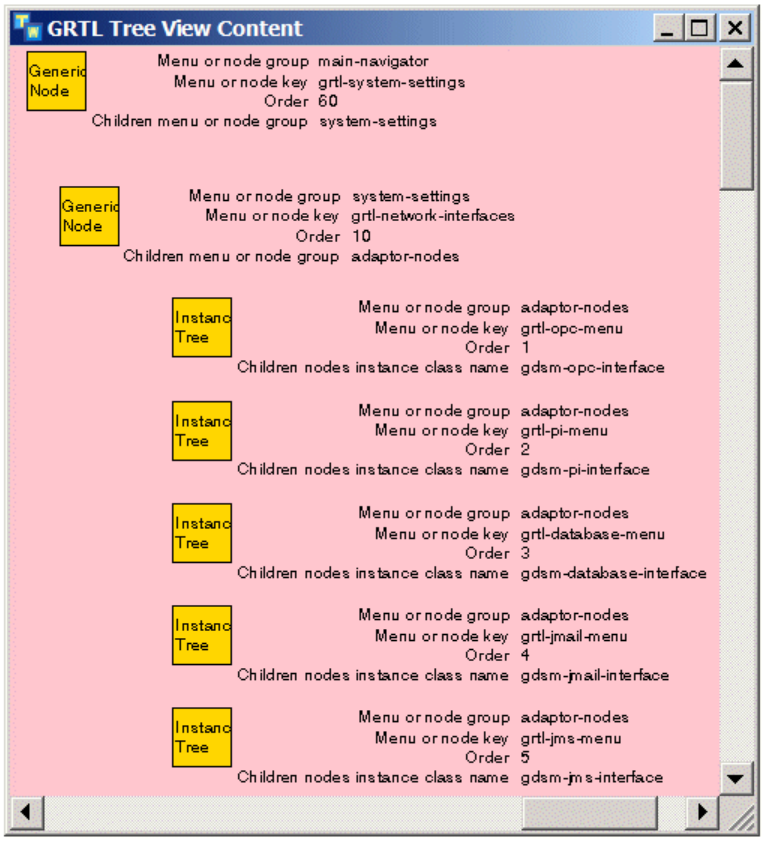
Access Tables	Message browser access tables. Toolboxes: none Views: properties
Templates	Message browser templates. Toolboxes: none Views: properties
Units	Engineering Unit Definitions.
Converter	Tool for converting units Toolboxes: none Views: properties
Conversions	Generic node grouping for Engineering unit conversion definitions. Toolboxes: none Views: properties
Synonyms	Generic node grouping for Engineering unit synonym definitions. Toolboxes: none Views: properties
Users	Generic node grouping for all users (user preferences). Toolboxes: none Views: properties
System Performance	System performance metrics Toolboxes: none Views: properties, metrics tabular view
Event & Alarms	Event and alarm metrics Toolboxes: none Views: properties, metrics in tabular view

Default Project Menu and Navigator Specification

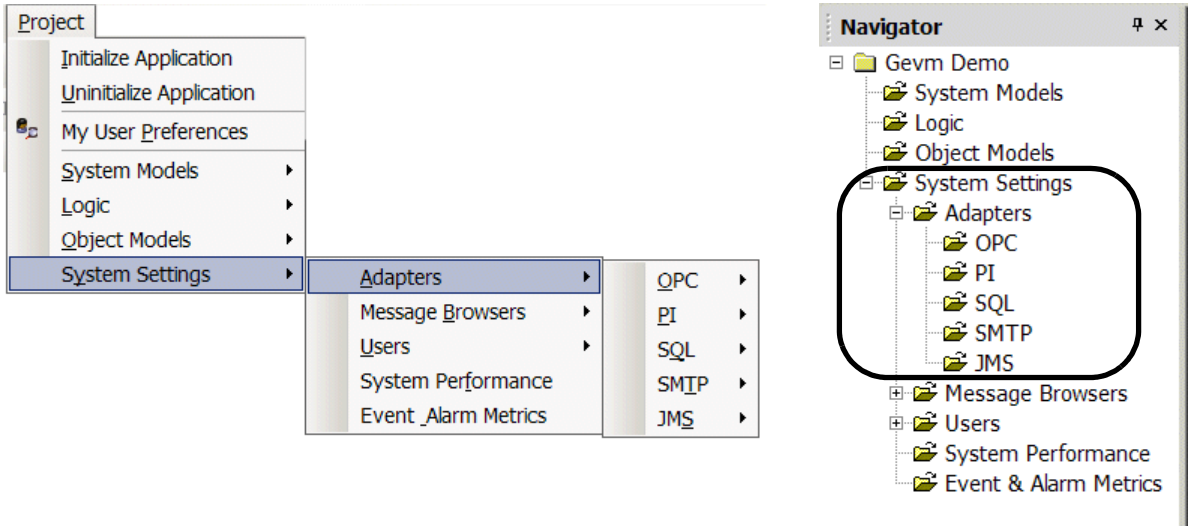
To access the default specification of the Project menu and Navigator, display the grtl-top-level workspace, click the Programmer's Interface button, click the Default Native Menubar, Toolbars, and Windows button, and click the Tree View Content button:



Here is a portion of the Tree View Content workspace, which shows the specification for the Adapter node and its subnodes in the Project menu and Navigator:



Here is the corresponding Project menu and Navigator:



Classes

grtl-node-specification

Generic abstract node specification.

Class Inheritance Path

grtl-node-specification, object, item

Attributes

Attribute	Description
menu-or-node-group	The group of the menu or node. Main-navigator represent the root entries of the tree or Project menu. This attribute is used to build groups of menus/tree nodes per level so that all matching names will end up at the same level.
Allowable values:	symbol
Default value:	main-navigator
text-localization-resource	The GFR localization resource.
Allowable values:	symbol
Default value:	grtl-text-localization-resource
menu-or-node-key	The unique key of the node or menu.
Allowable values:	symbol
Default value:	g2, with an index
menu-icon	An icon for the menu or node.
Allowable values:	symbol
Default value:	none
order	Within a group, nodes are sorted by the order attribute instead of alphabetically.
Allowable values:	integer
Default value:	1

enabled-if-modules-loaded	A list of G2 module names, as symbols. The tree node or menu entry is enabled and included if the specified modules are loaded.
Allowable values:	sequence
Default value:	sequence(the symbol grtl)
visible-in-user-modes	A list of symbolic names of user modes. the nodes is visible only in the specified user mode.
Allowable values:	sequence
Default value:	sequence()
visible-in-tree-view	Whether to show the node in the navigator tree. If false, the node is only visible in the Project menu.
Allowable values:	truth-value
Default value:	true
show-popup-menu-in-tree-view	If true, displays a popup menu on this node in tree view upon right click.
Allowable values:	truth-value
Default value:	false
user-data	User data for the node.
Allowable values:	sequence
Default value:	sequence()

grtl-generic-node-specification

A generic node specification that can have children nodes or menus.

Class Inheritance Path

grtl-generic-node-specification, grtl-node-specification, object, item

Attributes

Attribute	Description
menu-or-node-group	See grtl-node-specification.
Allowable values:	
Default value:	
text-localization-resource	See grtl-node-specification.
Allowable values:	
Default value:	
menu-or-node-key	See grtl-node-specification.
Allowable values:	
Default value:	
menu-icon	See grtl-node-specification.
Allowable values:	
Default value:	
order	See grtl-node-specification.
Allowable values:	
Default value:	
enabled-if-modules-loaded	See grtl-node-specification.
Allowable values:	
Default value:	

visible-in-user-modes See grtl-node-specification.

Allowable values:

Default value:

visible-in-tree-view See grtl-node-specification.

Allowable values:

Default value:

show-popup-menu-in-tree-view See grtl-node-specification.

Allowable values:

Default value:

user-data See grtl-node-specification.

Allowable values:

Default value:

children-menu-or-node-group The menu-or-node-group name of the children menus. It looks for grtl-node-specification instances with matching name in menu-or-node-group to collect the children.

Allowable values: symbol

Default value: none

get-menu-enabled-and-checked-status-procedure A procedure to check if the menu should be enabled and/or checked. The signature of procedure is:

```
my-menu-enabled-and-checked-procedure
   (spec: class grtl-generic-node-specification,
    win: class ui-client-item)
-> enabled: truth-value, checked: truth-value
```

Allowable values: symbol

Default value: unspecified

grtl-instance-hierarchy-node-specification

A node specification that displays an instance hierarchy as the children of the node or menu.

Class Inheritance Path

grtl-instance-hierarchy-node-specification, grtl-node-specification, object, item

Attributes

Attribute	Description
menu-or-node-group	See grtl-node-specification.
Allowable values:	
Default value:	
text-localization-resource	See grtl-node-specification.
Allowable values:	
Default value:	
menu-or-node-key	See grtl-node-specification.
Allowable values:	
Default value:	
menu-icon	See grtl-node-specification.
Allowable values:	
Default value:	
order	See grtl-node-specification.
Allowable values:	
Default value:	
enabled-if-modules-loaded	See grtl-node-specification.
Allowable values:	
Default value:	

visible-in-user-modes See grtl-node-specification.

Allowable values:

Default value:

visible-in-tree-view See grtl-node-specification.

Allowable values:

Default value:

show-popup-menu-in-tree-view See grtl-node-specification.

Allowable values:

Default value:

user-data See grtl-node-specification.

Allowable values:

Default value:

manage-dialog-title A symbol naming a key in a localization text file that is the name of the Manage dialog. The key is converted to a text if there is no matching key in the localization file.

Allowable values: symbol

Default value: none

children-nodes-instance-class-name Class name representing the type of object that will make up the child nodes for this grtl-instance-hierarchy-node-specification.

Allowable values: symbol

Default value: g2

children-nodes-exclude-instance-class-name Class name representing the class of objects that are excluded as child nodes, given children-nodes-instance-class-name, typically, a subclass of children-nodes-instance-class-name.

Allowable values: symbol

Default value: g2

organize-children-nodes-by-category

Whether to organize the associated children nodes by category. If true, the child nodes are sorted into categories, which are defined by grtl-get-category.

Allowable values: truth-value

Default value: false

manage-dialog-extra-classic-buttons

A sequence of symbols that represent extra buttons that are added into the associated manage dialog for the node in the classic user interface.

Allowable values: sequence

Default value: sequence()

manage-dialog-extra-native-buttons

A sequence of symbols that represent extra buttons that are added into the associated manage dialog for the node in the native user interface.

Allowable values: sequence

Default value: sequence()

new-instance-class-selection-procedure

Procedure called when the user selects New Instance on a node. The procedure represented by this symbol must have this signature:

```
my-callback
(spec: class grtl-instance-hierarchy-node-specification,
root-class-name: symbol, win: class ui-client-item)
-> selected-class-name: symbol
```

Allowable values: symbol

Default value: unspecified

new-instance-subworkspace-class-name

The class name of the subworkspace of the newly created instances.

Allowable values: symbol

Default value: none

name-new-instances

Whether to name new instances created when the user chooses New Instance on a node.

Allowable values: truth-value

Default value: false

instance-has-property-dialog Whether to allow property dialogs to be displayed from the Manage dialog for the node.

Allowable values: truth-value

Default value: true

show-subworkspace Whether to allow showing the subworkspace of the node.

Allowable values: truth-value

Default value: false

import-and-export Currently not implemented.

Allowable values: truth-value

Default value: false

grtl-class-hierarchy-node-specification

A node specification that displays a class hierarchy as the children of the node or menu.

Class Inheritance Path

grtl-class-hierarchy-node-specification, grtl-node-specification, object, item

Attributes

Attribute	Description
menu-or-node-group	See grtl-node-specification.
Allowable values:	
Default value:	
text-localization-resource	See grtl-node-specification.
Allowable values:	
Default value:	
menu-or-node-key	See grtl-node-specification.
Allowable values:	
Default value:	
menu-icon	See grtl-node-specification.
Allowable values:	
Default value:	
order	See grtl-node-specification.
Allowable values:	
Default value:	
enabled-if-modules-loaded	See grtl-node-specification.
Allowable values:	
Default value:	

visible-in-user-modes See grtl-node-specification.

Allowable values:

Default value:

visible-in-tree-view See grtl-node-specification.

Allowable values:

Default value:

show-popup-menu-in-tree-view See grtl-node-specification.

Allowable values:

Default value:

user-data See grtl-node-specification.

Allowable values:

Default value:

manage-dialog-title The title of the Manage dialog.

Allowable values: symbol

Default value: none

root-class-name The root class name.

Allowable values: symbol

Default value: item

new-instance-class-name The class name of instances of the class.

Allowable values: symbol

Default value: class-definition

grtl-module-hierarchy-node-specification

A node specification that displays the module hierarchy as the children of the node or menu.

Class Inheritance Path

grtl-module-hierarchy-node-specification, grtl-node-specification, object, item

Attributes

Attribute	Description
menu-or-node-group	See grtl-node-specification.
Allowable values:	
Default value:	
text-localization-resource	See grtl-node-specification.
Allowable values:	
Default value:	
menu-or-node-key	See grtl-node-specification.
Allowable values:	
Default value:	
menu-icon	See grtl-node-specification.
Allowable values:	
Default value:	
order	See grtl-node-specification.
Allowable values:	
Default value:	
enabled-if-modules-loaded	See grtl-node-specification.
Allowable values:	
Default value:	

visible-in-user-modes See grtl-node-specification.

Allowable values:

Default value:

visible-in-tree-view See grtl-node-specification.

Allowable values:

Default value:

show-popup-menu-in-tree-view See grtl-node-specification.

Allowable values:

Default value:

user-data See grtl-node-specification.

Allowable values:

Default value:

root-module-name The name of the root module.

Allowable values: symbol

Default value: unspecified

grtl-workspace-hierarchy-node-specification

A node specification that displays a workspace hierarchy as the children of the node or menu.

Class Inheritance Path

grtl-workspace-hierarchy-node-specification, grtl-node-specification, object, item

Attributes

Attribute	Description
menu-or-node-group	See grtl-node-specification.
Allowable values:	
Default value:	
text-localization-resource	See grtl-node-specification.
Allowable values:	
Default value:	
menu-or-node-key	See grtl-node-specification.
Allowable values:	
Default value:	
menu-icon	See grtl-node-specification.
Allowable values:	
Default value:	
order	See grtl-node-specification.
Allowable values:	
Default value:	
enabled-if-modules-loaded	See grtl-node-specification.
Allowable values:	
Default value:	

visible-in-user-modes See grtl-node-specification.

Allowable values:

Default value:

visible-in-tree-view See grtl-node-specification.

Allowable values:

Default value:

show-popup-menu-in-tree-view See grtl-node-specification.

Allowable values:

Default value:

user-data See grtl-node-specification.

Allowable values:

Default value:

root-workspace-name The name of the root workspace.

Allowable values: symbol

Default value: unspecified

grtl-separator-node-specification

A node specification that displays a separator in the Project menu.

Class Inheritance Path

grtl-workspace-hierarchy-node-specification, grtl-node-specification, object, item

Procedures

grtl-add-menus-for-menu-specifications
(*menu-or-node-group*: symbol, *parent-menu*: item-or-value,
menu-commands: class hash-table, *win*: class ui-client-item)

Adds submenus based on specifications of grtl-node-specification instances.

grtl-get-tree-view-hierarchy-for-specifications
(*child-window*: class grtl-child-window, *handle*: integer, *win*: class g2-window)
-> structure

Collects a tree view specification based on specifications of grtl-node-specification instances.

24

Palettes

Describes a class to create and organize palettes, as well as procedures to manipulate the palettes.

Introduction

GRTL provides a palette group class, grtl-palette-group, for organizing palettes into groups. Palette group instances provide information about the palette such as its name and associate group and the content of the palette is stored on its subworkspace. Items with GFR palette behavior on its subworkspace are considered entries and clonable items for a palette group. For information on adding palette behavior to items, see the *G2 Foundation Resources User's Guide*.

GRTL automatically displays palettes using a shortcut bar or listbar if the native user interface is supported. The content of these native palettes is based on grtl-palette-group instances. The procedure _grtl-get-toolbox-folders collects palette groups and items with GFR palette behavior on the subworkspace of these palette groups to build the folders and folder content.

GRTL also provides procedures for constructing palette group menus from a set of GMS palettes and showing palettes in palette group menus and submenus. You use the APIs that construct palette groups and panels as part of the callback for a GMS dynamic cascade template callback.

For general information on GMS palettes and dynamic cascade templates, see the *G2 Menu System User's Guide*.

Example

You use the grtl-construct-palette-group and grtl-construct-palette-panel APIs as part of the callbacks for a GMS dynamic cascade template callback to dynamically build menus that contain palettes.

Here is an example of how you would build a dynamic palette menu. The example builds a submenu with the union of all palette groups and a submenu of each palette group with the specific palettes.

First, you create a GMS dynamic cascade template with the following configuration. Notice that the gms-subpanel-constructor specifies the callback procedure named gdev-construct-panel-group-panel.

a gms-dynamic-cascade-template

UUID	"936794900dfe11d7b931000802639ab8"
Notes	OK
Names	none
Gms restricted modes	a symbol-array
Gms index	51
Gms user key	none
Gms text resource group	none
Gms label	gdev-palettes-menu
Gms help label	gdev-palettes-menu-help
Gms activation callback	gdev-palette-selection-callback
Gms inline icon class	none
Gms inline icon description	none
Gms initially enabled	true
Gms native icon	unspecified
Gms selection callback	none
Gms posting callback	none
Gms preconstruct panel	false
Gms subpanel constructor	gdev-construct-palette-group-panel
Gms reclaim templates	true

Here is the callback procedure that dynamically builds the palette menu group by calling the grtl-construct-palette-group API. This example filters out any grtl-palette-group instances whose palette-group is "Intelligent Objects". It also sorts the list of grtl-palette-group names, based on the group-order attribute of the grtl-palette-group objects, instead of sorting them alphabetically, the default. The second argument to the API procedure, gdev-construct-palette-panel, is the name of a procedure that is called when the user selects the palette group name, which builds a submenu of the individual palettes within each group.

```
gdev-construct-palette-group-panel (Handle: integer, ActivationInfo: item-or-value,
    Index: integer, TemplatesList: class item-list)
win: class ui-client-item;
groups-to-exclude: sequence = sequence ("Intelligent Objects");
begin
    win = call gms-get-window-for-handle (Handle);
    call grtl-construct-palette-group(TemplatesList, the symbol
```

```
        gdev-construct-palette-panel, groups-to-exclude, false, Win);
end
```

Here is the gdev-construct-palette-panel callback procedure, which calls grtl-construct-palette-panel to dynamically build the list of palettes, based on the palette-name attribute of each grtl-palette-group instance for each palette group.

```
gdev-construct-palette-panel (Handle: integer, ActivationInfo: item-or-value,
    Index: integer, TemplatesList: class item-list)
group: value;
win: class ui-client-item;
palettesToExclude: sequence = sequence ( ) ;
begin
    win = call gms-get-window-for-handle (Handle);
    group = call gms-get-activation-property (ActivationInfo, Index, the symbol
        gms-user-key );
    call grtl-construct-palette-panel(TemplatesList, group, palettesToExclude,
        true, Win);
end
```

When the user selects a palette name, GMS calls the procedure specified in the gms-activation-callback specified in the gms-dynamic-cascading-template. In this example, it calls gdev-palette-selection-callback, which calls grtl-show-palette to display the subworkspace of the appropriate grtl-palette-group instance.

```
gdev-palette-selection-callback (MenuHandle: integer, ActivationInfo: item-or-value,
    Index: integer)
GroupText, NameText: text;
Win: class g2-window;
begin
    Win = call gms-get-window-for-handle (MenuHandle);
    GroupText = call gms-get-activation-property (ActivationInfo, Index, the symbol
        gms-user-key);
    NameText = call gms-get-activation-property (ActivationInfo, Index, the symbol
        gms-label);
    call grtl-show-palette(GroupText, NameText, Win);
end
```

Classes

grtl-palette-group

grtl-palette-group

Provides a palette group on whose subworkspace you create palettes. To implement the palette behavior, use the GFR palette facility to add palette behavior to items on the subworkspace of grtl-palette-group instances.

You create one palette group for each palette in the group, where each palette group has the same palette group name. You can override the default palette group order and palette order, in which case each palette group must have the same order.

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

Class Inheritance Path

grtl-palette-group, grtl-object, object, grtl-item, item

Attributes

Attribute	Description
palette-group	The name of the palette group. For example, the palette group named Custom Domain Objects might appear in a menu sequence such as this: View > Palettes > Custom Domain Objects.
Allowable values:	Any text
Default value:	""
palette-name	The name of the palette within the palette group. For example, the palette named Custom Datapoints might appear in a menu sequence such as this: View > Palettes > Custom Domain Objects > Custom Datapoints.
Allowable values:	Any text
Default value:	""
group-order	The sort order for the palette group. By default, the palette group is sorted alphabetically. To override this default, specify the group-order as an integer, where 0 is the first palette group, 1 is the next, and so on.
Allowable values:	Any integer

Default value: 100

palette-order The sort order for the palette within the group. By default, the palettes within the group are sorted alphabetically. To override this default, specify the palette-order as an integer, where 0 is the first palette, 1 is the next, and so on.

Allowable values: Any integer

Default value: 0

icon-name The name of a user-defined or built-in class used as the icon for the palette.

Allowable values: symbol

Default value: none

valid-user-modes A list of symbols indicating modes in which palette is visible.

Allowable values: sequence

Default value: sequence()

valid-workspace-classnames Used by a procedure to dynamically enable or disable palette folders, based on the active window. This attribute lists class names for which this palette would be enabled when a workspace of any of these classes is in the active child window. The _grtl-enable-disable-toolbox-folders procedure enables and disables folders, based on this attribute.

Allowable values: sequence

Default value: sequence()

valid-superior-classnames Used by a procedure to dynamically enable or disable palette folders, based on the active window. This attribute lists class names of items superior to the active workspace for which palette would be enabled. The _grtl-enable-disable-toolbox-folders procedure enables and disables folders, based on this setting.

Allowable values: sequence

Default value: sequence()

show-in-native-ui If true, displays palette in native user interface; otherwise, the palette is not shown.

Allowable values: symbol

Default value: true

Procedures

grtl-construct-palette-group

Synopsis

grtl-construct-palette-group
(*templates-list*: item-list , *callback*: symbol , *groups-to-exclude*: sequence, *sort-by-name*: truth-value , *client*: ui-client-item)

Argument	Description
templates-list	An item list of GMS menu objects to create dynamically. See the description of dynamic menus in the *G2 Menu System User's Guide*.
callback	A symbol naming a callback procedure to execute when the user selects the palette group name in a GMS menu.
groups-to-exclude	A sequence of strings that name grtl-palette-group instances to exclude from the palette group.
sort-by-name	True to sort the palette group alphabetically by name, or false to sort the palette groups, based on the group-order attribute of the grtl-palette-group instances.
client	The client, typically a G2 window.

Description

Used in conjunction GMS menus to dynamically create the menu entries referring to palettes. See Example.

grtl-construct-palette-panel

Synopsis

grtl-construct-palette-panel
(*templates-list*: item-list , *palette-group*: text , *palettes-to-exclude*: sequence, *sort-by-name*: truth-value , *client*: ui-client-item)

Argument	Description
templates-list	An item list of GMS menu objects to create dynamically. See the description of dynamic menus in the *G2 Menu System User's Guide*.
palette-group	The display name of the palette group that contains the individual palettes, typically, the gms-user-key of the GMS dynamic cascade template.
palettes-to-exclude	A sequence of strings that name grtl-palette-group instances to exclude from the palette panel.
sort-by-name	True to sort the palettes within the panel alphabetically by name, or false to sort the palettes, based on the palette-order attribute of the grtl-palette-group instances.
client	The client, typically a G2 window.

Description

Used in conjunction with GMS menus to dynamically create menu entries that display individual palettes within a palette group. See Example.

grtl-show-palette

Synopsis

grtl-show-palette
(*group-name*: text, *palette-name*: text, *client*: ui-client-item)

Argument	Description
group-name	The display name of the palette group that contains the individual palettes, for instance, the gms-user-key of the GMS dynamic cascade template.
palette-name	The display name of the palette, for instance, the gms-label of the GMS dynamic cascade template.
client	The client, typically a G2 window.

Description

Used in conjunction with GMS menus to display a palette given its group and name.

25

Child Windows and Panes

Describes procedures to manage child windows and panes.

Introduction

Child windows and panes are typically used to display user interface elements for easy navigation, organization of the application, or to enhance the user experience. Examples of child window panes are navigator tree views, palettes, alarm notification panels, or performance dashboards. Pane window names are typically also added to the View menu to enable users to show or hide them, as needed.

Most Telewindows child windows are displayed in either an MDI child container or in a pane window. MDI child windows can be resized, tiled, cascaded, or iconified, whereas pane windows can be floating or docked on any side of the Telewindows window, and auto-hidden when not used.

The following window types are supported, all of which except the workspace view can be displayed in an MDI child container, in a pane container, or in a dialog container:

- Tree view.
- HTML view.
- Shortcut bar.
- Dialog view.
- Chart view.
- Workspace view, which can only be displayed in a MDI child container.

Note that you can also create workspace views by using standard G2 syntax or system procedures, for example, using the show action, the Go To Subworkspace menu choice, and the g2-ui-show-workspace system procedure.

A child window is created by creating an instance of a grtl-child-window. The window-name attribute specifies the name of the child window and should be unique. The populate-procedure of the grtl-child-window instance refers to the procedure that provides the content for the view. Other attributes on this object provide other specifications for the view.

If a pane container is selected for the child window, the localized name of the child window appears in the View menu in the menu bar. Selecting the name of the child window shows or hides the pane window.

To provide localization, GRTL collects additional information regarding the child window from the GRTL localization resource. To get the label of the title, which is also displayed in the View menu, use the following syntax, where *window-name* is the name assigned to the child window:

CHILD-WINDOW.*window-name*.TITLE, "My Pane View"

Configuring Child Windows on Startup

You can configure the attributes of child windows in the configuration file, *config.txt*, by default.

To configure child windows, the section name in the configuration file must match the name of the child window. Under that section, configure the following attributes of the child window:

- enabled
- initially-visible
- window-priority
- state

- large-icon-size
- closeable
- minimizeable
- maximizeable
- resizeable
- floatable
- autohideable
- draggable
- initial-dock
- neighbor-dock
- neighbor-window-name
- left
- top
- width
- height

grtl-child-window

Class Inheritance Path

grtl-child-window, object

Attributes

Attribute	Description
window-class	The type of child window to display, as a symbol.
Allowable values:	g2-tree-view, g2-html-view, g2-dialog-view, g2-chart-view, g2-shortcut-bar, g2-workspace-view
Default value:	g2-tree-view
window-name	The name of the child window, which must be unique.
Allowable values:	symbol
Default value:	g2
enabled	Whether the child window is enabled.
Allowable values:	truth-value
Default value:	true
window-priority	Upon connection, all visible child windows are sorted by their window-priority and displayed in this order, starting from small numbers.
Allowable values:	integer
Default value:	0
initially-visible	If true, the child window is displayed upon connection.

Attribute	Description
Allowable values:	truth-value
Default value:	false
visible-in-user-modes	A sequence of symbols that specify the user modes in which the child window is visible.
Allowable values:	sequence
Default value:	sequence()
caption-is-visible	If true, displays the caption of the window.
Allowable values:	truth-value
Default value:	true
caption-title	The text of the caption.
Allowable values:	text
Default value:	""
icon-name	The name of a G2 class displayed as an icon in the caption bar.
Allowable values:	symbol
Default value:	none
state	The state of the window, as a symbol.
Allowable values:	normal, minimized, maximized, docked, hidden, autohidden, autoshown, closed
Default value:	normal

Attribute	Description
container	Specifies the type of container, as a symbol. If displayed as a pane, the window-name is added to the View menu to show and hide the pane.
Allowable values:	mdi-child, pane
Default value:	pane
style	Currently only used if the window class is g2-shortcut-bar and specifies the display style.
Allowable values:	default, listbar
Default value:	default
closeable	If true, displays a close box in the upper-right corner of the window.
Allowable values:	truth-value
Default value:	true
minimizeable	If true, displays the minimize button in the upper-right corner of the window. This attribute is only relevant if container is mdi-child.
Allowable values:	truth-value
Default value:	true
maximizeable	If true, displays the maximize button in the window. This attribute is only relevant if container is mdi-child.
Allowable values:	truth-value
Default value:	true
resizeable	If true, enables resizing of the window.

Attribute	Description
Allowable values:	truth-value
Default value:	true
floatable	If true, enables the user to undock a pane to make it floating. This attribute is only relevant if container is pane.
Allowable values:	truth-value
Default value:	true
autohideable	If true, displays the user interface widget to auto hide the window. This attribute is only relevant if container is pane.
Allowable values:	truth-value
Default value:	true
draggable	If true, enables users to drag panes from on docking position to another. This attribute is only relevant if container is pane.
Allowable values:	truth-value
Default value:	true
initial-dock	Specifies the initial docking mode of the window. This attribute is only relevant if container is pane.
Allowable values:	top, bottom, left, right, float
Default value:	left
neighbor-window-name	The name of the neighbor window. Upon display of a child window, the window is positioned relative to the neighbor window, if visible. This attribute is only relevant if container is pane.

Attribute	Description
Allowable values:	symbol
Default value:	none
neighbor-dock	Specifies the docking mode relative to its neighbor, if the neighbor window is visible. This attribute is only relevant if container is pane.
Allowable values:	left, right, top, bottom, float, within
Default value:	within
left	Specifies the left position of the child window. This attribute is only relevant for floating panes and mdi-child containers.
Allowable values:	integer (positive)
Default value:	0
top	Specifies the top position of the child window. This attribute is only relevant for floating panes and mdi-child containers.
Allowable values:	integer (positive)
Default value:	0
width	Specifies the width of the child window.
Allowable values:	integer (positive)
Default value:	240
height	Specifies the height of the child window. This attribute is only relevant for floating panes and mdi-child containers.
Allowable values:	integer (positive)

Attribute	Description
Default value:	350
custom-background-color	The background color for the child window.
Allowable values:	symbol
Default value:	unspecified
custom-foreground-color	The foreground color for the child window.
Allowable values:	symbol
Default value:	unspecified

Attribute	Description
populate-procedure	Specifies the procedure to call to collect the content to display in the child window. GRTL comes with several predefined procedures described later in this chapter. The signature of this procedure is: my-populate-procedure (*child-window*: class grtl-child-window, *handle*: integer, *win*: class g2-window) -> *return-value*: value The *return-value* depends to the window class, as follows: • g2-tree-view — Returns a structure describing the tree. See g2-ui-populate-tree-view in the *G2 System Procedures Reference Manual*. • g2-shortcut-bar — Returns a sequence of structures describing the folders. See g2-ui-create-shortcut-bar in the *G2 System Procedures Reference Manual*. • g2-html-view — Returns the URL to display in the HTML view, as a text. • g2-dialog-view — Returns a structure describing a custom dialog specification. See g2-ui-post-custom-dialog in the *G2 System Procedures Reference Manual*. • g2-chart-view — Returns a structure describing the chart view. See g2-ui-modify-chart-view in the *G2 System Procedures Reference Manual*. • g2-workspace-view — Currently not used. GRTL provides several built-in populate procedures. See Built-in Populate Procedures. *Allowable values:* symbol *Default value:* unspecified

Attribute	Description
update-procedure	Specifies a procedure that returns the content for updating the child window. The signature of this procedure is the same as populate-procedure. *Allowable values:* symbol *Default value:* unspecified
callback-procedure	Specifies a procedure that is called upon user interaction with the child window, such as selecting an node in a tree view or selecting an item in a shortcut bar. The custom procedure should also call the default procedure to ensure, for example, that window configurations are saved and restored next time the child window is displayed. The signature of this procedure is: my-child-window-callback (*event*: symbol, *win*: class g2-window, *handle*: integer, *key*: value, *info*: structure, *user-data*: value) *Allowable values:* symbol *Default value:* unspecified
selection-update-procedure	Specifies a procedure that is called when the selection changes. For example, you can use this procedure to enable or disable folders in a shortcut bar, based on the active view. The signature of this procedure is: selection-procedure (*child-window*: class grtl-child-window, *selected-wksp*: item-or-value, *selected-items*: sequence, *win*: class g2-window) *Allowable values:* symbol *Default value:* unspecified

Attribute	Description
content-argument	Storage that can be used by the named procedures for any purpose.
Allowable values:	value
Default value:	none
large-icon-size	When true, uses large icons in shortcut bars. Set to false to use small icons.
Allowable values:	truth-value
Default value:	true
min-width	Specify the minimal allowed width of the child window.
Allowable values:	integer (positive)
Default value:	0
min-height	Specify the minimal allowed height of the child window.
Allowable values:	integer (positive)
Default value:	0

Built-in Populate Procedures

GRTL provides the following built-in procedures for the populate-procedure attribute. Many of these are used internally by GRTL for the default user interface.

_grtl-get-url
(*child-window*: class grtl-child-window, *handle*: integer, *win*: class g2-window) -> *url*: text

If the content-argument of the child window contains a text, returns it as the URL; otherwise, returns the text specified in the application-default-url of the active grtl-module-settings. This predefined procedure is suitable for the g2-tree-view window class.

_grtl-get-module-hierarchy
(*child-window*: class grtl-child-window, *handle*: integer, *win*: class g2-window) -> *module-hierarchy*: structure

Returns the module hierarchy. This procedure is suitable for the g2-tree-view window class.

_grtl-get-class-hierarchy
(*child-window*: class grtl-child-window, *handle*: integer, *win*: class g2-window) -> *class-hierarchy*: structure

Returns the class hierarchy of item or the root class name specified in the content-argument of the child window. This procedure is suitable for the g2-tree-view window class.

grtl-get-tree-view-hierarchy-for-specifications
(*child-window*: class grtl-child-window, *handle*: integer, *win*: class g2-window) -> *project-hierarchy*: structure

Returns the project application navigator hierarchy, based on a specification of grtl-node-specification instances. This procedure is suitable for the g2-tree-view window class.

_grtl-get-palette
(*child-window*: class grtl-child-window, *handle*: integer, *win*: class g2-window) -> *folders*: sequence

Returns palette folders for the palette-group specified in the content-argument of the child window. It returns one folder for each grtl-palette-group with a matching palette-group value. Each folder contains an entry for every item with gfr-palette behavior that exists on the subworkspace of the selected grtl-palette-group instance. This procedure is suitable for the g2-shortcut-bar window class.

_grtl-get-toolbox-folders
(*child-window*: class grtl-child-window, *handle*: integer, *win*: class g2-window)
-> *folders*: sequence

Similar to _grtl-get-palette but returns one folder for every grtl-palette-group, which means all palette groups appear. This procedure is suitable for the g2-shortcut-bar window class and is typically used with the listbar style.

_grtl-enable-disable-toolbox-folders
(*child-window*: class grtl-child-window, *selected-wks*: item-or-value, *selected-items*: sequence, *win*: class g2-window)

A procedure that can be specified as the selection-update-procedure of a grtl-child-window used as a palette to enable and disable folders, based on the active window. See grtl-palette-group for the context used to enable or disable palette groups. This procedure is suitable for the g2-shortcut-bar window class with the style set to default.

Additional Procedures

Below is a list of procedures, but typically developers do not need to call them because the GRTL user interface environment, if enabled, calls them to configure the user interface upon connection.

grtl-create-child-windows
(*user*: class grtl-user-preferences, *session*: class grtl-user-session, *win*: class ui-client-item)

Creates instances of every grtl-child-window with initially-visible set to true for the specified user window. Displays the window in the user modes specified in visible-in-user-modes.

grtl-delete-child-windows
(*user*: class grtl-user-preferences, *session*: class grtl-user-session, *win*: item-or-value)

Deletes all grtl-child-window instances visible in the specified user window.

grtl-show-hide-or-update-child-window
(*child-window*: class grtl-child-window, *destroy-existing-views*: truth-value, *option*: value, *win*: class g2-window)

Shows or hides specific a grtl-child-window in the specified user window. The *option* argument is only used for g2-html-view. It it is a symbol, it is sent to the view by calling g2-ui-manage-html-view (*option*, *handle*, false, *win*), which enables you to send commands such as refresh and home to the HTML view. If the *destroy-existing-view* argument is true and the window is visible, the window is destroyed; otherwise, it is refreshed.

grtl-is-child-window-visible
(*child-window*: class grtl-child-window, *win*: class g2-window)
-> *visible*: truth-value, *handle*: integer

Returns true and the window handle if the child window is visible in the specified G2 window.

grtl-show-url
(*url*: text, *win*: class g2-window)

Displays the specified URL in the default GRTL built-in HTML child window.

26

Dialogs

Describes general procedures related to dialogs, which provide an abstract API that displays a Microsoft dialog or a UIL dialog, depending on the system and the user configuration.

Introduction

GRTL and many other developer modules include native dialogs for their classes. You build these dialogs, using the functionality provided by the GDU module. The dialogs are displayed when selecting the properties menu or when calling the following API:

grtl-show-properties
(target: class item, Win: class g2-window)
-> *return-value*: truth-value

Returns true if the dialog is accepted, false if cancelled.

Note that almost all implementations of grtl-show-properties for Gensym dialog classes display a native dialog if supported by Telewindows; otherwise, a UIL dialog is displayed, or the G2 table is displayed if no dialog has been defined. When developing custom dialog classes, we recommend that you do the same by calling grtl-use-native-user-interface to determine which dialog to display at runtime.

We recommend implementing the grtl-show-properties method for your custom classes to ensure smooth integration within the user interface framework. This method is called independently of the origin of the request to display the properties dialog and, therefore, works as an abstraction layer. Selecting the Properties menu choice in the menu bar, in the toolbar, in a popup menu, or by pressing F4 are some of the actions that display the properties dialog of an item.

In addition to providing these property dialogs for classes defined in the developer module, GRTL includes several convenience APIs to either display UIL dialogs or native dialogs. The *message* or *caption* argument of many of these APIs are specified as a value, in which case the argument can be a symbol or a text. If specified as a symbol, the text is localized by calling grtl-localize-message-without-substitutions.

grtl-confirmation-dialog

Synopsis

grtl-confirmation-dialog
(*message*: value , *caption*: value , *win*: class g2-window)
-> *button*: symbol

Argument	Description
message	The message to display in the dialog, which can be a symbol or a text.
caption	The caption to display in the title bar of the dialog, which can be a symbol or a text.
win	The window in which to display the dialog.

Return Value	Description
button	The button selected by the user, as a symbol, which can be the symbol yes or cancel.

Description

Creates and posts a confirmation dialog with a message, caption, and OK and Cancel button.

grtl-error-dialog

Synopsis

grtl-error-dialog
(*message*: value , *caption*: value , *win*: class g2-window)
-> *button*: symbol

Argument	Description
message	The message to display in the dialog, which can be a symbol or a text.
caption	The caption to display in the title bar of the dialog, which can be a symbol or a text.
win	The window in which to display the dialog.

Return Value	Description
button	The button selected by the user, as a symbol, which can be the symbol ok or cancel.

Description

Creates and posts an error dialog with a message, caption, error icon, and OK and Cancel button.

grtl-information-dialog

Synopsis

grtl-information-dialog
    (*message*: value , *caption*: value , *win*: class g2-window)
    -> *button*: symbol

Argument	Description
message	The message to display in the dialog, which can be a symbol or a text.
caption	The caption to display in the title bar of the dialog, which can be a symbol or a text.
win	The window in which to display the dialog.

Return Value	Description
button	The button selected by the user, as a symbol, which can be the symbol ok or cancel.

Description

Creates and posts an information dialog with a message, caption, information icon, and OK and Cancel button.

grtl-post-delay-notification

Synopsis

grtl-post-delay-notification
(*message*: text, *caption*: text, *client*: ui-client-item)

Argument	Description
message	The delay notification message.
caption	The text to display in the caption of the notification dialog.
client	The client, typically a G2 window.

Description

Posts a new delay notification message in a Windows dialog, if available in the client; otherwise, calls grtl-post-notification in a G2 classic client. See grtl-post-notification.

grtl-post-delete-confirmation-dialog

Synopsis

grtl-post-delete-confirmation-dialog
(*client*: g2-window, *item-or-msg*: item-or-value, *list-inteface*: truth-value)
-> *response*: symbol

Argument	Description
client	The client requesting this operation
item-or-msg	The item to delete or the message to post
list-inteface	If true, confirmation dialog should offer extra options 'yes to all' and 'cancel'

Return Value	Description
response	The selected button: yes-to-all, no, or cancel.

Description

Posts a dialog requesting confirmation to delete one or more items and returns the user response. This procedure displays a Windows dialog, if available in the client; otherwise, displays a UIL confirmation dialog in a G2 classic client.

grtl-post-generic-dialog

Synopsis

grtl-post-generic-dialog
(*message*: text, *message-type*: symbol, *font-size*: symbol, *client*: g2-window, *icon*: symbol)
-> *button-label*: text, *return-value*: item-or-value

Argument	Description
message	The text to post in the dialog.
message-type	The type of generic dialog to use. Specify confirmation, message, notification, or query.
font-size	The size of the text. Specify small, medium, or large.
client	The client, typically a G2 window.
icon	The class-name of an icon to display with text. UIL provides the following icons: uil-information-icon, uil-question-icon, uil-warning-icon.

Return Values	Description
button-label	The label of the button selected: Yes, No, OK, or Cancel. Query dialogs return Yes (if the OK button is selected) or No (if the Cancel button is selected.)
return-value	The value returned by a query dialog, or "".

Description

This API extends the uil-post-generic-dialog API to post a generic Windows dialog, if available in the client; otherwise, posts a UIL dialog in a G2 classic client.

If *message-type* is message and *icon* is uil-warning-icon, the procedure displays a warning dialog. If *message-type* is message and *icon* is uil-information-icon, the procedure displays an information dialog.

For details, see Chapter 6 "Dialogs" in the *G2 GUIDE/UIL Procedures Reference Manual*.

grtl-post-progress-notification

Synopsis

grtl-post-progress-notification
    (*message*: text, *caption*: text, *client*: ui-client-item)

Argument	Description
message	The progress notification message.
caption	The text to display in the caption of the notification dialog.
client	The client, typically a G2 window.

Description

Posts a progress notification message in a Windows dialog, if available in the client; otherwise, calls grtl-post-notification in a G2 classic client. See grtl-post-notification.

grtl-post-simple-message

Synopsis

grtl-post-simple-message
    (*text-or-resource*: value, *icon-name*: symbol, *client*: class ui-client-item)

Argument	Description
text-or-resource	A text or a valid text resource, as a symbol, for the current language.
icon-name	The symbol uil-information-icon or uil-warning-icon.
client	The ui-client-item for posting the dialog.

Description

Creates and posts a simple dialog containing some text, an icon, and an OK button.

grtl-print-workspace

Synopsis

grtl-print-workspace
    (*wksp*: kb-workspace, *win*: g2-window)

Argument	Description
wksp	The workspace to print.
win	The window containing the workspace.

Description

Displays a Windows print dialog, if available in the window; otherwise, displays a UIL print dialog in a G2 classic client.

grtl-remove-delay-notification

Synopsis

grtl-remove-delay-notification
    (*client*: ui-client-item)

Argument	Description
client	The client, typically a G2 window.

Description

Removes the progress notification message in the client.

grtl-remove-progress-notification

Synopsis

grtl-remove-progress-notification
    (*client*: ui-client-item)

Argument	Description
client	The client, typically a G2 window.

Description

Removes the progress notification message in the client.

grtl-retry-cancel-dialog

Synopsis

grtl-retry-cancel-dialog
(*message*: value, *caption*: value, *win*: class g2-window)
-> *button*: symbol

Argument	Description
message	The message to display in the dialog, which can be a symbol or a text.
caption	The caption to display in the title bar of the dialog, which can be a symbol or a text.
win	The window in which to display the dialog.

Return Value	Description
button	The button selected by the user, as a symbol, which can be the symbol retry or cancel.

Description

Creates and posts a confirmation dialog with a message, caption, and Retry and Cancel buttons.

grtl-update-progress-notification

Synopsis

grtl-update-progress-notification
(*message*: text, *client*: ui-client-item)

Argument	Description
message	The updated text.
client	The client, typically a G2 window.

Description

Updates the progress notification message in the Windows progress notification dialog, if available in the client; otherwise calls grtl-post-notification in a G2 classic client. See grtl-post-notification.

grtl-warning-dialog

Synopsis

grtl-warning-dialog
    (*message*: value , *caption*: value , *win*: class g2-window)
    -> *button*: symbol

Argument	Description
message	The message to display in the dialog, which can be a symbol or a text.
caption	The caption to display in the title bar of the dialog, which can be a symbol or a text.
win	The window in which to display the dialog.

Return Value	Description
button	The button selected by the user, as a symbol, which can be the symbol ok or cancel.

Description

Creates and posts a warning dialog with a message, caption, warning icon, and OK and Cancel button.

grtl-yes-no-dialog

Synopsis

grtl-yes-no-dialog
(*message*: value, *caption*: value, *win*: class g2-window)
-> *button*: symbol

Argument	Description
message	The message to display in the dialog, which can be a symbol or a text.
caption	The caption to display in the title bar of the dialog, which can be a symbol or a text.
win	The window in which to display the dialog.

Return Value	Description
button	The button selected by the user, as a symbol, which can be the symbol yes or no.

Description

Creates and posts a confirmation dialog with a message, caption, and Yes and No buttons.

grtl-yes-no-cancel-dialog

Synopsis

grtl-yes-no-cancel-dialog
(*message*: value, *caption*: value, *win*: class g2-window)
-> *button*: symbol

Argument	Description
message	The message to display in the dialog, which can be a symbol or a text.
caption	The caption to display in the title bar of the dialog, which can be a symbol or a text.
win	The window in which to display the dialog.

Return Value	Description
button	The button selected by the user, as a symbol, which can be the symbol yes, no, or cancel.

Description

Creates and posts a confirmation dialog with a message, caption, and Yes, No, and Cancel buttons.

Audit Trail

Describes how to enable the audit trail.

Introduction

GRTL supports creating an audit trail of user login and log out activities. This feature might be useful for analyzing usage patterns, including how best to scale applications and servers beyond the basic audit trail capabilities.

You can track user activities in a log file. The file is formatted as a comma separated value (CSV) to make it easy to load into Excel.

To enable the audit trail:

1. Create an instance of a grtl-module-settings copying it from the current active one and store it in any KB module higher in the module hierarchy.
2. Change the user-audit-file of the grtl-module-settings created in step 1 to your desired log file name.
3. Change the enable-user-audit-file of the grtl-module-settings created in step 1 to true.
4. Restart G2.

Part IV

Object Model

Chapter 28: General Object Model

Describes core classes and methods that subclasses can customize, as well as procedures for creating, deleting and cloning items, which call the item constructore and destructors, as appropriate.

Chapter 29: Unique Identifier Keys

Describes an implementation of a universal name/ID system.

Chapter 30: Properties

Describes methods to manipulate properties of a G2 object model.

Chapter 31: Event Notification

Describes the object model event notification, which is based on the G2 event notification model.

Chapter 32: User Interface

Describes methods to support user interface operations of the object model.

Chapter 33: Instance Repository

Describes the instance repository, which is a standard mechanism for organizing newly created instances without requiring the user to know where objects are stored.

Chapter 34: Datapoints

Describes datapoint classes and methods, including classes and methods to represent datapoints in process maps.

Chapter 35: Process Maps, Domain Objects, and Schemes

Describes definitions and operations for process maps, domain objects, and schemes.

Chapter 36: XML Serialization

Describes methods for XML serialization of G2 objects.

28

General Object Model

Describes core classes and methods that subclasses can customize, as well as procedures for creating, deleting and cloning items, which call the item constructore and destructors, as appropriate.

Introduction

GRTL defines a root item and object class, which define default implementations of various methods for the item class. These methods include a constructor method, a destructor method, an initialization method, and a method for copying all public attributes from a source to a destination object. Other methods determine if the item is clonable, selectable, resizable, and movable, and whether the item can be renamed and deleted. GRTL provides default implementations of these methods for the item class.

GRTL also provides procedures for cloning, creating, and deleting GRTL items, which automatically call the GRTL constructor and destructor for the item, as appropriate. These procedures also notify listeners of the appropriate type of event if the item is also a subclass of grtl-event-source.

Classes

grtl-item

The root GRTL item class.

Class Inheritance Path

grtl-item, item

grtl-object

The root GRTL object class.

Class Inheritance Path

grtl-object, object, grtl-item, item

Methods and Procedures

Methods

item::grtl-add-item-to-project-tree-view
item::grtl-can-delete
item::grtl-can-rename
item::grtl-constructor
item::grtl-copy-attributes
item::grtl-destructor
item::grtl-get-category
item::grtl-get-hostname-or-ip-address
item::grtl-get-system-info-or-best-practice-url
item::grtl-get-uuid
item::grtl-initialize
item::grtl-is-clonable
item::grtl-is-item-private

Procedures

grtl-clone
grtl-create
grtl-delete

item::grtl-add-item-to-project-tree-view

Synopsis

item::grtl-add-item-to-project-tree-view
    (*itm*: class item)
    -> *return-value*: truth-value

Argument	Description
itm	The item to add.

Return Value	Description
return-value	True if the item should be added to the project tree, false otherwise.

Description

As items on a workspace are added to the tree view, for example, this method is called for each instance to determine if it should be included in the project tree or not.

item::grtl-can-delete

Synopsis

item::grtl-can-delete
    (*itm*: item, *client*: ui-client-item)
    -> *delete*: truth-value

Argument	Description
itm	The item to test.
client	The client, typically a G2 window.

Return Value	Description
delete	true if the item can be deleted, false otherwise.

Description

Determines if an item can be deleted or cut. The default implementation for the item class returns true.

item::grtl-can-rename

Synopsis

item::grtl-can-rename
    (*itm*: item, *client*: ui-client-item)
    -> *rename*: truth-value

Argument	Description
itm	The item to test.
client	The client, typically a G2 window.

Return Value	Description
rename	True if the item can be renamed, false otherwise.

Description

Determines if an item can be renamed. The default implementation for the item class returns true.

item::grtl-constructor

Synopsis

item::grtl-constructor
    (*itm*: item, *clone-source*: item-or-value)

Argument	Description
itm	The item to initialize.
clone-source	If cloned from another item, the source item to clone or the symbol none.

Description

Provides the default implementation of the constructor for instances of the class item. If the item instance is cloned from another item, *clone-source* refers to the source item. If the item is newly created, *clone-source* is the symbol none. The default implementation of the constructor for the item class does nothing.

The reason to use the same constructor for new and cloned items is to avoid overhead and to provide easy synchronization. Otherwise, you would have to call the default constructor to create the item, then call another API to clone the item. By combining them, you avoid unnecessary initialization overhead during basic construction and it is easier to keep track of what needs to be initialized or overwritten if the item is cloned from another one.

item::grtl-copy-attributes

Synopsis

item::grtl-copy-attributes
    (*source*: item, *destination*: item, *notify*: truth-value)

Argument	Description
source	The source item.
destination	The destination item.
notify	True to notify listeners of the destination item.

Description

Copies specified attributes from the source object to the destination object. The default implementation for the item class copies all public attributes from the source object to the destination object. It uses the APIs grtl-get-properties, grtl-get-values-of-properties, and grtl-set-values-of-properties to perform the copy. For more information, see Properties.

item::grtl-destructor

Synopsis

item::grtl-destructor
    (*itm*: item)

Argument	Description
itm	The item to delete.

Description

Provides the default implementation of the destructor for instances of the class item. The default implementation does nothing.

item::grtl-get-category

Synopsis

item::grtl-get-category
    (*itm*: item)
    -> *category*: text

Argument	Description
itm	The item whose category to get.

Return Value	Description
category	The category of the item.

Description

Returns the category of an item, for example, the category of an event detection diagram or a fault model. As items organized by category are added to the project tree, for example, this method is called for each instance to collect the category names.

item::grtl-get-hostname-or-ip-address

Synopsis

item::grtl-get-hostname-or-ip-address
    (*itm*: class item)
    -> *info*: text

Argument	Description
itm	The item whose info to get.

Argument	Description
info	The network host name or IP address of the host associated with the target item

Description

Returns the network host name or IP address of the host associated with the target item. Returning an empty string means that no host name or IP address is available. For example, you can use this API to launch a Telnet session or ping the remote host.

item::grtl-get-system-info-or-best-practice-url

Synopsis

item::grtl-get-system-info-or-best-practice-url
(*itm*: class item)
-> *info*: text

Argument	Description
itm	The item whose info to get.

Return Value	Description
info	The system info or best practice URL of the target item.

Description

Returns the system-information-or-best-practice-url of a grtl-domain-object. Returning an empty string means that none exits.

item::grtl-get-uuid

Synopsis

item::grtl-get-uuid
    (*itm*: item)
    -> *uuid*: text

Argument	Description
itm	The item whose UUID to get.

Return Value	Description
uuid	The uuid of the item

Description

Returns the uuid of an item as a text.

item::grtl-initialize

Synopsis

item::grtl-initialize
    (*itm*: item)

Argument	Description
itm	The item to initialize.

Description

Initializes or re-initializes the item.

item::grtl-is-clonable

Synopsis

item::grtl-is-clonable
    (*itm*: item, *client*: ui-client-item)
    -> *clonable*: truth-value

Argument	Description
itm	The item to test.
client	The client, typically a G2 window.

Return Value	Description
clonable	true if the item can be cloned or copied, false otherwise.

Description

Determines if an item can be cloned or copied. The default implementation for the item class returns true.

item::grtl-is-item-private

Synopsis

item::grtl-is-item-private
    (*itm*: item)
    -> *private*: truth-value

Argument	Description
itm	The item to test.

Return Value	Description
private	True if the item is private, false otherwise.

Description

Determines if an item is public or private.

grtl-clone

Synopsis

grtl-clone
    (*source*: item)
    -> *new-object*: item-or-value

Argument	Description
source	The item to clone.

Return Value	Description
new-object	The new object.

Description

Clones an existing object. It also ensures that the grtl-constructor API is called for the newly created object. If the class is a subclass of grtl-event-source, after calling the constructor, it automatically notifies any listeners with an event type of cloned. See grtl-event-source.

grtl-create

Synopsis

grtl-create
    (*class-name*: symbol)
    -> *new-item*: item-or-value

Argument	Description
class-name	The class name of the item to create.

Return Value	Description
new-item	The newly created item.

Description

Creates a new instance of the specified class. If class is a subclass of grtl-object, it automatically calls the grtl-constructor method. If the class is a subclass of grtl-event-source, after calling the constructor, it automatically notifies any listeners with an event type of created. See grtl-event-source.

grtl-delete

Synopsis

grtl-delete
    (*itm*: item)

Argument	Description
itm	The item to delete.

Description

Deletes an item. If class is a subclass of grtl-object, it automatically calls the grtl-destructor method. If the class is a subclass of grtl-event-source, after calling the constructor, it automatically notifies any listeners with an event type of deleted. See grtl-event-source. This procedure also removes any operator message associated with the deleted item.

29

Unique Identifier Keys

Describes an implementation of a universal name/ID system.

Introduction

GRTL provides a way of implementing a universal name/ID system that allows different classes to use different attributes for a name/ID. Keys are human readable and should be unique for a class. You can create a unique key that is application-specific, such as a network-managed domain object reference, which is different from the standard G2 UUID. An analogy is the unique key of database tables.

The advantages of this system are:

- Keys can be human readable.
- Keys can be defined and set at the application level.
- You can still use a symbol attribute as the key for a class.
- A pre-existing class can be made to support the key API without modification, provided it has an existing attribute that serves the purpose. The default implementation uses the UUID as the key.

The disadvantages of this system are:

- You must implement methods for each base class that you want to support keys.
- The implementation is a little slower than using pure symbols; therefore, be sure to use indexed attributes as keys.

Classes

grtl-item-with-key

The core GRTL class that defines an item with a key.

Class Inheritance Path

grtl-item-with-key, grtl-item, item

Attributes

Attribute	Description
key	A text string that is meaningful for a human reader and unique for the item's base class.
Allowable values:	Any text
Default value:	""

Methods

grtl-item-with-key::grtl-get-key
grtl-item-with-key::grtl-get-key-attribute-name
grtl-item-with-key::grtl-set-key

grtl-object-with-key

The core GRTL class that defines an object with a key.

Class Inheritance Path

grtl-object-with-key, object, grtl-item-with-key, grtl-item, item

Attributes

Attribute	Description
key	See grtl-item-with-key.
Allowable values:	inherited
Default value:	""

Methods

grtl-object-with-key::cdg-get-name-as-string

Methods

grtl-class-definition::grtl-get-key-attribute-name

Synopsis

grtl-class-definition::grtl-get-key-attribute-name
(*item-class*: grtl-class-definition)
-> *key*: symbol

Argument	Description
item-class	The class whose key attribute name to get.

Return Value	Description
key	The key attribute name.

Description

Returns the key attribute name of the specified class. See grtl-class-definition.

class-definition::grtl-get-item-by-key

Synopsis

class-definition::grtl-get-item-by-key
(*item-class*: class-definition, *key*: text, *hierarchy-containment*: item-or-value)
-> *itm*: class item

Argument	Description
item-class	The class of the item whose key to get.
key	The key of the item get.
hierarchy-containment	A workspace or item that needs to be in the containment hierarchy of the selected item, or false if none.

Return Value	Description
itm	The item with the specified key.

Description

Returns the item with the specified key and of the specified class. If no instance with a matching key is found, the key attribute is converted to a G2 object name to determine if a matching named object exists.

The *hierarchy-containment* argument can be false, or it can be an item or a workspace. If false, no containment test is performed. If the argument is an item or a workspace, then in addition to selecting objects with the correct key, this method selects objects where the containment argument is a member of the containment hierarchy of the selected object. For example, you can use this technique to find objects that are within a model or model detail, when other models containing objects with the same key may exist as well.

class-definition::grtl-get-item-by-key-with-existence-check

Synopsis

class-definition::grtl-get-item-by-key-with-existence-check
(*item-class*: class-definition, *key*: text, *hierarchy-containment*: item-or-value)
-> *exists*: truth-value, *itm*: item-or-value

Argument	Description
item-class	The class of the item whose key to get.
key	The key of the item get.
hierarchy-containment	A workspace or item that needs to be in the containment hierarchy of the selected item, or false if none.

Return Value	Description
exists	True if the item exists, or false if the item does not exist.
itm	The item with the specified key.

Description

Determines if an item with a particular key exists, and, if found, returns true and the item; otherwise, returns false. If no instance with matching key is found, the key attribute is converted to a G2 object name to determine if a matching named object exists. See class-definition::grtl-get-item-by-key for a description of the containment argument.

class-definition::grtl-get-key-attribute-name

Synopsis

class-definition::grtl-get-key-attribute-name
(*item-class*: class-definition)
-> *attribute-name*: symbol

Argument	Description
item-class	The class of the item whose key attribute name to get.

Return Value	Description
attribute-name	The key attribute name.

Description

Returns the key attribute name of the specified item class.

class-definition::grtl-item-with-key-exists

Synopsis

class-definition::grtl-item-with-key-exists
(*item-class*: class-definition, *key*: text, *hierarchy-containment*: item-or-value)
-> *exists*: truth-value

Argument	Description
item-class	The class of the item whose key to get.
key	The key of the item get.
hierarchy-containment	A workspace or item that needs to be in the containment hierarchy of the selected item, or false if none.

Return Value	Description
exists	True if the item exists, or false if the item does not exist.

Description

Determines if an item with a particular key exists, and, if found, returns true; otherwise, returns false. If no instance with matching key is found, the key attribute is converted to a G2 object name to determine if a matching named object exists. See class-definition::grtl-get-item-by-key for a description of the containment argument.

grtl-item-with-key::grtl-get-key

Synopsis

grtl-item-with-key::grtl-get-key
    (*itm*: grtl-item-with-key)
    -> *key*: text

Argument	Description
itm	The item whose key to get.

Return Value	Description
key	The key of the specified item.

Description

Returns the key of the specified item. Use the methods grtl-item-with-key-exists and grtl-get-item-by-key to search the items of a particular class, where an attribute is defined that is used as the key. The default implementation of this method for the grtl-item-with-key class signals an error when the method has not been implemented for the class of the specified item.

grtl-item-with-key::grtl-get-key-attribute-name

Synopsis

grtl-item-with-key::grtl-get-key-attribute-name
    (*itm*: grtl-item-with-key)
    -> *attribute-name*: symbol

Argument	Description
itm	The item whose key to get.

Return Value	Description
attribute-name	The key attribute name.

Description

Returns the key attribute name of the specified item.

grtl-item-with-key::grtl-set-key

Synopsis

grtl-item-with-key::grtl-set-key
    (*itm*: grtl-item-with-key, *key*: text)

Argument	Description
itm	The item whose key to set.
key	The new value for the key.

Description

Sets the key of the specified item.

item::grtl-get-item-by-key

Synopsis

item::grtl-get-item-by-key
(*item-class*: item, *key*: text, *hierarchy-containment*: item-or-value)
-> *itm*: class item

Argument	Description
item-class	The class of the item whose key to get.
key	The key of the item to get.
hierarchy-containment	A workspace or item that needs to be in the containment hierarchy of the selected item, or false if none.

Return Value	Description
itm	The item with the specified key.

Description

Determines if an item with a particular key exists, and, if found, returns true and the item; otherwise, returns false. If no instance with matching key is found, the key attribute is converted to a G2 object name to determine if a matching named object exists. See class-definition::grtl-get-item-by-key for a description of the containment argument.

The default implementation of this method for the item class calls grtl-get-item-with-uuid-of-class to find objects, based on the G2 UUID used as a key. Your implementations should only search among instances of the specified class and should signal the error grtl-item-not-found if the item could not be found.

item::grtl-get-item-by-key-with-existence-check

Synopsis

item::grtl-get-item-by-key-with-existence-check
(*itm*: item, *key*: text, *hierarchy-containment*: item-or-value)
-> *exists*: truth-value, *itm*: item-or-value

Argument	Description
item-class	The class of the item whose key to get.
key	The key of the item to get.
hierarchy-containment	A workspace or item that needs to be in the containment hierarchy of the selected item, or false if none.

Return Value	Description
exists	True if the item exists, or false if the item does not exist.
itm	The item with the specified key.

Description

Determines if an item with a particular key exists, and, if found, returns true and the item; otherwise, returns false. If no instance with matching key is found, the key attribute is converted to a G2 object name to determine if a matching named object exists. See class-definition::grtl-get-item-by-key for a description of the containment argument.

item::grtl-get-key

Synopsis

item::grtl-get-key
    (*itm*: item)
    -> *key*: text

Argument	Description
itm	The item whose key to get.

Return Value	Description
key	The key of the specified item.

Description

Returns the key of the specified item. Use the methods grtl-item-with-key-exists and grtl-get-item-by-key to search the items of a particular class, where an attribute is defined that is used as the key. The default implementation of this method for the item class signals an error when the method has not been implemented for the class of the specified item.

item::grtl-get-key-attribute-name

Synopsis

item::grtl-get-key-attribute-name
    (*itm*: item)
    -> *attribute-name*: symbol

Argument	Description
itm	The item whose key to get.

Return Value	Description
attribute-name	The key attribute name.

Description

Returns the key attribute name of the specified item.

item::grtl-item-with-key-exists

Synopsis

item::grtl-item-with-key-exists
(*item-class*: item, *key*: text, *hierarchy-containment*: item-or-value)
-> *exists*: truth-value

Argument	Description
item-class	The class of the item whose key to get.
key	The key of the item get.
hierarchy-containment	A workspace or item that needs to be in the containment hierarchy of the selected item, or false if none.

Return Value	Description
exists	True if the item exists, or false if the item does not exist.

Description

Determines if an item with a particular key exists, and, if found, returns true; otherwise, returns false. If no instance with matching key is found, the key attribute is converted to a G2 object name to determine if a matching named object exists. See class-definition::grtl-get-item-by-key for a description of the containment argument.

The default implementation of this method for the item class matches the key against the G2 UUID of the object. The default implementation signals an error if the method has not been implemented for the specified class. Your implementations of this method should accept an instance of your class as the first argument and only search among instances of that class, where an attribute is defined that is used as the key.

Note that performance is better if you call the custom implementations directly; however, to do that, you must have an instance of the class at the time of the call.

item::grtl-set-key

Synopsis

item::grtl-set-key
    (*itm*: item, *key*: text)

Argument	Description
itm	The item whose key is to be set.
key	The item's new key

Description

Sets the key of the item specified. The default implementation for the item class unconditionally signals an error if this method has not been implemented for the class of the specified item.

30

Properties

Describes methods to manipulate properties of a G2 object model.

Introduction

GRTL provides a class definition that you can use to define classes with properties, which can be virtual attributes or can map directly to G2 attributes. Properties help to separate the programmer model from the way properties are exposed to user interfaces and applications. Properties extend the G2 attribute type definition by providing an object model to define information such as units, extended data types, constraints such as minimum and maximum values, prompt label, text field that should only contain filenames, and so on. Therefore, you can define and set additional specifications, for example, to build XML schemas, SQL statements, or dialogs. These specifications are defined on the server and are used and propagated to clients, which helps maintain consistency and manage properties.

GRTL provides methods and procedures to get and set property information of an item, as well as procedures for getting property information for all items and classes.

Classes

grtl-class-definition

A class definition with extra attributes to store a class description and extended object model/property information. The format of the property definition is identical to the format returned by item::grtl-get-property-attributes, except that it enables you to define the properties at the class level instead of defining custom methods for each class. This makes it much easier to edit properties in dialogs, for example.

Class Inheritance Path

grtl-class-definition, class-definition, definition, item

Attributes

Attribute	Description
class-description	A description of the class.
Allowable values:	Any text
Default value:	""
instance-key-attribute-name	The attribute name of instances of the class that is used as the key.
Allowable values:	Any symbol
Default value:	UUID

class-attribute-properties A sequence of structures that describes an extended object model for properties. See item::grtl-get-properties for details of the content of each structure. Each structure should contain at least the following information:

structure
(PROPERTY-NAME: symbol, {property name}
DESCRIPTION: TEXT, {description of property}
LABEL: text, {default label to use in dialogs/reports}
LABEL-KEY: text, {any text without spaces used as a key to locate the lable in multi-language systems}
HELP-KEY: text, {any key used by help system to locate online help}
TYPE: symbol, {see below for valid list of choices and specs}
PUBLIC: truth-value {true if property is public, false if property is private}
READ-ONLY: truth-value, {true if it is a read-only attribute, false otherwise}
CATEGORY: text {category of property used by property browser to group properties}
MIN: quantity {minimum value for quantity values or minimum string length for texts}
MAX: quantity {maximum value for quantity values or maximum string length for texts}
PRECISION: float {decimal point precision, where 0 means all}
UNITS: text {units}
CHOICES: sequence () | symbol {fixed list of choices as a sequence or a procedure name as a symbol}

Allowable values: Any sequence

Default value: sequence ()

Methods and Procedures

Methods

item::grtl-confirmed-set-value
item::grtl-get-item-properties
item::grtl-get-properties
item::grtl-get-properties-type-info
item::grtl-get-property-type-info
item::grtl-get-property-value
item::grtl-get-units
item::grtl-get-value
item::grtl-get-values-of-properties
item::grtl-get-values-of-properties-as-structure
item::grtl-set-property-value
item::grtl-set-property-value
item::grtl-set-property-value
item::grtl-set-property-value
item::grtl-set-property-value
item::grtl-set-value
item::grtl-set-value-with-collection-time
item::grtl-set-values-of-properties
item::grtl-set-values-of-properties-as-structure

Procedures

grtl-get-properties-for-class

item::grtl-confirmed-set-value

Synopsis

item::grtl-confirmed-set-value
(*itm*: item, *new-value*: value, *timeout*: integer)

Argument	Description
itm	The item to update.
new-value	The new value.
timeout	The number of milliseconds to wait for confirmation.

Description

Updates the value of an item, checks that the value has been correctly set, and if not, throws a grtl-timed-out-awaiting-set-confirmation error. By default, this method sets the value of a variable or parameter. Otherwise, you can provide your own implementation that sets the value of a particular attribute of an item.

item::grtl-get-item-properties

Synopsis

item::grtl-get-item-properties
(*itm*: item, *user-mode*: text)
-> *properties*: structure

Argument	Description
itm	The item whose properties to get.
user-mode	The user mode in which the properties are visible for the item.

Return Value	Description
properties	A structure containing the properties of the specified item, as well as other information.

Description

Returns information about the properties of an item in a given user mode. The procedure returns a structure with the following syntax:

structure
(CLASS-NAME: symbol, {the class name as a symbol}
UUID: text, {the uuid of the item}
KEY-NAME: symbol, {the item key property name}
KEY-VALUE: value, {the item key property value}
PROPERTIES: sequence {sequence of structures listing all properties with detailed information })

For information on the properties structure, see item::grtl-get-property-type-info.

item::grtl-get-properties

Synopsis

item::grtl-get-properties
(*itm*: item, *settable*: truth-value, *user-mode*: text)
-> *properties*: structure

Argument	Description
itm	The item whose properties to get.
settable	True to get all properties; false to get read-only properties only.
user-mode	The user mode in which the properties are visible for the item.

Return Value	Description
properties	A structure containing the properties of the specified item, as well as other information.

Description

Returns information about the public properties of an item in a given user mode. You can get all properties or only read-only properties. The procedure returns a structure with the following syntax:

```
structure
(CLASS-NAME: symbol, {the class name as a symbol}
UUID: text, {the uuid of the item}
KEY-NAME: symbol, {the item key property name}
KEY-VALUE: value, {the item key property value}
PROPERTIES: sequence {sequence of structures listing all properties with
      detailed information } )
```

For information on the properties structure, see item::grtl-get-property-type-info.

The default implementation of this method returns all user-defined attributes that are not proprietary, in addition to several G2 hidden system attributes. The code calls grtl-get-property-type-info to determine if properties are private and settable. The list of hidden G2 system attributes that are automatically added is: notes, names, item-width, item-height, item-x-position, and item-y-position.

item::grtl-get-properties-type-info

Synopsis

item::grtl-get-properties-type-info
(*itm*: item, *user-mode*: text)
-> *properties*: sequence

Argument	Description
itm	The item whose properties information to get.
user-mode	The user mode in which the properties are visible for the item.

Return Value	Description
properties	A sequence of structures of all property attributes of the specified item.

Description

Returns the property attributes of all attributes of the specified item.

For information on the structure, see item::grtl-get-property-type-info.

item::grtl-get-property-type-info

Synopsis

item::grtl-get-property-type-info
(*itm*: item, *property-name*: symbol)
-> *property-info*: structure

Argument	Description
itm	The item whose property information to get.
property-name	The property name of the item whose property information to get.

Return Value	Description
property-info	The attributes of the specified property and item.

Description

Returns extended property information. G2 does not support a way to specify extended attribute properties such as minimum, maximum, precision, and read-only properties. Therefore, you can use this API to query such information. Defining this information on the server ensures a unique central definition. User interfaces, dialogs and any other clients can use this API to retrieve extended specifications.

The basic implementation attempts to determine the properties, based on the attribute type. In some cases, however, the G2 data type may not be the desired specification, such as with durations. Thus, subclasses can overwrite this method.

Instances of variable-or-parameter support the following extended property attributes:

- value: The current value of the instance. It supports querying the property info, getting the value, and setting the initial value.
- initial-value: The initial value of the instance. It supports querying the property info, and getting and setting the value.
- history: The history values of the instance. It supports querying the property info, and getting and setting the value.

Instances of grtl-datapoint support the following extended property attribute, in addition to those for variable-or-parameter:

- units: The units of the instance. It supports querying the property info, and getting and setting the value.

Instances of grtl-derived-datapoint support the following extended property attributes, in addition to those for grtl-datapoint:

- formula: The formula of the instance. A formula is a mathematical expression that can refer to other datapoints. The result of the formula defines the value of the derived datapoint instance. It supports querying the property info, and getting and setting the value.

Below is a description of the structure that this method should return. For a description of each property, see Property Constraints and Information.

```
structure
(PROPERTY-NAME: symbol, {property name}
DESCRIPTION: TEXT, {description of
      property}
LABEL: text, {default label to use in
      dialogs/reports}
LABEL-KEY: text, {any text without spaces
      used as a key to locate the lable in
      multi-language systems}
HELP-KEY: text, {any key used by help
      system to locate online help}
TYPE: symbol, {see below for valid list of
      choices and specs}
PUBLIC: truth-value {true if property is public,
      false if property is private}
READ-ONLY: truth-value, {true if it is a
      read-only attribute, false otherwise}
CATEGORY: text {category of property used
      by property browser to group properties}
MIN: quantity {minimum value for quantity
      values or minimum string length for texts}
MAX: quantity {maximum value for quantity
     values or maximum string length for texts}
PRECISION: float {decimal point precision,
     where 0 means all}
UNITS: text {units}
CHOICES: symbol | sequence ( ) {fixed list of
      choices as a sequence or a procedrue name
      as a symbol}
```

The signature of the procedure specified in choices should have the following signature and should return a list of valid values, which is used to present the list of choices in a combo box:

```
my-choices-proc (itm: class my-class, win: class ui-client-item) = (sequence)
```

You can specify the type to be more specific than the built-in G2 types. For example, you can use G2 strings to specify dates, durations, filenames, URLs, and so on. By adding a more specific type specification, you can use the generic G2 data types without having to create subclasses. You can then use the type specification when configuring the UI. For example, if you specify file as the type when the G2 attribute type is a text, the user interface can build a control for a text field with a button to open a file browser. Similarly, you can use an untyped G2 attribute and specify duration as the type to store only durations.

The following are valid values for type in the structure. Each type lists the corresponding G2 attribute type, default XML mapping, and default SQL mapping. For a description of each value type, see Property Types.

- text: any text
- symbol: any G2 valid symbol
- truth-value: a boolean value
- integer: a positive or negative integer
- float: a positive or negative floating point number
- quantity: a positive or negative number
- currency: a positive or negative currency number
- datetime: G2 date-time format (number relative to start of G2 process)
- duration: G2 duration format (number that can be specified G2 format:2 weeks, 3 days and 2 hours)
- iso-time: time in ISO 8601 time format: 15:45:00.000
- iso-datetime: date-time in ISO 8601 datetime format: 1998-07-12:16:30:00.000
- iso-duration: duration in ISO 8601 duration format: P30D
- iso-date: date in ISO 8601 date format: 05-07-1995
- iso-month: month in ISO 8601 Gergorian month format: --07--
- iso-year: year in ISO 8601 Gregorian year format: 1998, 2002
- iso-yearmonth: year and month in ISO 8601 Gregorian year month format: 1998-07
- iso-day: day in ISO 8601 Gregorian day format: ---12

- iso-monthday: month and day in ISO 8601 Gregorian month day format: --07-12
- color: G2 color as a symbol
- TableSequence: a table (spreadsheet) of x/y values. The data are stored in a sequence of rows, where each row is a sequence of values (a sequence of sequences).
- TableStructure: a table (spreadsheet) of x/y values. The data are stored in a sequence of rows, where each row is a structure of values (a sequence of structures). Each structure represents a row with the attributes of the structure representing the columns and containing the values of the cells.
- file: a string that refers to a file (relative or full pathname)
- html: a string that contains html code
- uri or url: a string that refers to a valid Uniform Resource Identifier
- sequence: a sequence of item or values

item::grtl-get-property-value

Synopsis

item::grtl-get-property-value
(*itm*: item, *property-name*: symbol)
-> *property-value*: item-or-value, *collection-time*: quantity

Argument	Description
itm	The item whose property value to get.
property-name	The property name of the item whose value to get.

Return Value	Description
property-value	The property value of the specified item.
collection-time	The collection time of the value.

Description

Returns the value of a property.

item::grtl-get-units

Synopsis

item::grtl-get-units
    (*itm*: item, *propertyname*: symbol)
    -> *units*: text

Argument	Description
itm	The item whose property to get.
property-name	The property name of the item whose property to get.

Return Value	Description
text	The units.

Description

Get the units of an item.

item::grtl-get-value

Synopsis

item::grtl-get-value
    (*itm*: item)
    -> *current-value*: value, *collection-time*: quantity

Argument	Description
itm	The item whose value to get.

Return Value	Description
current-value	The current value.
collection-time	The collection time of the value.

Description

Returns the current value of an item. By default, this method gets the value of a variable or parameter. Otherwise, you can provide your own implementation that gets the value of a particular attribute of an item.

item::grtl-get-values-of-properties

Synopsis

item::grtl-get-values-of-properties
(*itm*: item, *property-names*: sequence)
-> *property-values*: sequence

Argument	Description
itm	The item whose property values to get.
property-names	A sequence of symbols of property names whose values to get.

Return Value	Description
property-values	A sequence of values for the specified properties.

Description

Returns the values of a set of specified properties.

item::grtl-get-values-of-properties-as-structure

Synopsis

item::grtl-get-values-of-properties-as-structure
    (*itm*: item, *property-names*: sequence)
    -> *property-values*: structure

Argument	Description
itm	The item whose property values to get.
property-names	A sequence of symbols of property names whose values to get.

Return Value	Description
property-values	The values of the properties.

Description

Returns the values of a set of specified properties as a structure. The returned structure contains the values for each valid property. If a property does not exist for the item, no entry is added to the structure, but an error is dispatched via gerr-dispatch.

item::grtl-set-property-value

Synopsis

item::grtl-set-property-value
(*itm*: item, *property-name*: symbol, *new-value*: item-or-value)

Argument	Description
itm	The item whose property values to set.
property-name	The property name to set.
new-value	The new property value.

Description

Sets a property to a new value. The property must be a subclass of variable-or-parameter. This method relies on property attributes to manage the attribute, for example, to ensure that the value is within a valid range. You can use this method to set values from user interfaces or external systems to ensure the value is validated before storing it.

item::grtl-set-property-value

Synopsis

item::grtl-set-property-value
(*itm*: item, *property-name*: symbol, *new-value*: item-or-value, *collection-time*: quantity)

Argument	Description
itm	The item whose property values to set.
property-name	The property name to set.
new-value	The new property value.
collection-time	The collection time.

Description

Sets a property to a new value with a collection time, if applicable. The property must be a subclass of variable-or-parameter. This method relies on property attributes to manage the attribute, for example, to ensure that the value is within a valid range. You can use this method to set values from user interfaces or external systems to ensure the value is validated before storing it.

item::grtl-set-property-value

Synopsis

item::grtl-set-property-value
(*itm*: item, *property-name*: symbol, *new-value*: item-or-value, *apply-time-unit*: truth-value, *time-unit*: item-or-value)

Argument	Description
itm	The item whose property values to set.
property-name	The property name to set.
new-value	The new property value.
apply-time-unit	If the *property-name* is a duration type, and *apply-time-unit* is true, the *time-unit* conversion factor is multiplied by the *new-value*. See item::grtl-get-property-type-info.
time-unit	Time unit scale.

Description

Sets a property to a new value with a time unit, if applicable. The property must be a subclass of variable-or-parameter. This method relies on property attributes to manage the attribute, for example, to ensure that the value is within a valid range. You can use this method to set values from user interfaces or external systems to ensure the value is validated before storing it.

item::grtl-set-property-value

Synopsis

item::grtl-set-property-value
(*itm*: item, *property-name*: symbol, *new-value*: item-or-value,
apply-time-unit: truth-value, *time-unit*: item-or-value, *collection-time*: quantity)

Argument	Description
itm	The item whose property values to set.
property-name	The property name to set.
new-value	The new property value.
apply-time-unit	If the *property-name* is a duration type, and *apply-time-unit* is true, the *time-unit* conversion factor is multiplied by the *new-value*. See item::grtl-get-property-type-info.
time-unit	Time unit scale.
collection-time	The collection time.

Description

Sets a property to a new value with a time unit and collection time, if applicable. The property must be a subclass of variable-or-parameter. This method relies on property attributes to manage the attribute, for example, to ensure that the value is within a valid range. You can use this method to set values from user interfaces or external systems to ensure the value is validated before storing it.

item::grtl-set-property-value

Synopsis

item::grtl-set-property-value
(*itm*: item, *property-name*: symbol, *new-value*: item-or-value, *apply-time-unit*: truth-value, *time-unit*: item-or-value, *collection-time*: quantity, *ignore-read-only*: truth-value)

Argument	Description
itm	The item whose property values to set.
property-name	The property name to set.
new-value	The new property value.
apply-time-unit	If the *property-name* is a duration type, and *apply-time-unit* is true, the *time-unit* conversion factor is multiplied by the *new-value*. See item::grtl-get-property-type-info.
time-unit	Time unit scale.
collection-time	The collection time.
ignore-read-only	True to set property value even if it is read only.

Description

Sets a property to a new value with a time unit and collection time, if applicable, including read-only attributes. The property must be a subclass of variable-or-parameter. This method relies on property attributes to manage the attribute, for example, to ensure that the value is within a valid range. You can use this method to set values from user interfaces or external systems to ensure the value is validated before storing it.

item::grtl-set-value

Synopsis

item::grtl-set-value
    (*itm*: item, *new-value*: value)

Argument	Description
itm	The item whose value to set.
new-value	The new value.

Description

Sets the value of an item. By default, this method sets the value of a variable or parameter. Otherwise, you can provide your own implementation that sets the value of a particular attribute of an item.

item::grtl-set-value-with-collection-time

Synopsis

item::grtl-set-value-with-collection-time
(*itm*: item, *new-value*: value, *collection-time*: quantity)

Argument	Description
itm	The item whose value to set.
new-value	The new value.
collection-time	The collection time of the value.

Description

Sets the value of an item with a collection time, if applicable.

item::grtl-set-values-of-properties

Synopsis

item::grtl-set-values-of-properties
(*itm*: item, *property-names*: sequence, *property-values*: sequence)

Argument	Description
item	The item whose property values to set.
property-names	A sequence of symbols of property names whose values to set.
property-values	A sequence of corresponding property values.

Description

Sets a selected set of property values of an item. This method relies on property attributes to manage attributes, for example, to ensure that the value is within a valid range. Use this method to set values from user interfaces or external systems to ensure the value is validated before storing it.

item::grtl-set-values-of-properties-as-structure

Synopsis

item::grtl-set-values-of-properties-as-structure
(*itm*: item, *property-values*: structure)

Argument	Description
itm	The item whose property values to set.
property-values	The property values to set as a structure.

Description

Sets the values of a set of specified properties. This method relies on property attributes to manage attributes, for example, by ensuring that the value is within a valid range. Use this method to set values from user interfaces or external systems to ensure the value is validated before storing it. Note that passing entries in the structure for properties on the item that do not exist dispatches an error for logging or displayed using gerr-dispatch, but it does not abort the code to set all properties.

grtl-get-properties-for-class

Synopsis

grtl-get-properties-for-class
(*class-or-class-name*: item-or-value, *user-mode*: text)
-> *properties*: structure

Argument	Description
class-or-class-name	The class definition or class name whose properties to get.
user-mode	The user mode in which the properties are visible for the class.

Return Value	Description
properties	A structure containing the properties of the specified class, as well as other information.

Description

Returns information about the properties of the instances of a class in a given user mode, where *class-or-class-name* can either be a class definition or the name of the class as a text or symbol. The procedure returns a structure with the following syntax:

```
structure
(CLASS-NAME: symbol, {the class name as a symbol}
UUID: text, {the uuid of the item}
KEY-NAME: symbol, {the item key property name}
KEY-VALUE: value, {the item key property value}
PROPERTIES: sequence {sequence of structures listing all properties with
    detailed information } )
```

For information on the properties structure, see item::grtl-get-property-type-info.

31

Event Notification

Describes the object model event notification, which is based on the G2 event notification model.

Introduction

GRTL provides object model event notification, which is based on the G2 publish/subscribe event notification model. It is similar to those found in other languages such as .NET and Java, though compared to many other engines and since this implementation is tightly integrated with the engine, it ensure greater management. It also supports out-of-the-box distributed notification for G2-G2, G2-GSI, G2-Javalink, and G2-ActiveXLink providing a rich environment to build distributed applications. The GRTL object model event notification is object-oriented and enables listeners to register at the class level. Listeners registered at the class level are notified of changes of any instance, which simplifies the system setup.

GRTL provides an event source class, which defines methods for adding and removing event listeners, and dispatching events. It also provides an event source class definition, which allows you to define an event source at the class level, along with corresponding methods for adding and removing event listeners for all instances of the class.

GRTL also provides default implementations of methods for enabling and disabling event notification, and dispatching events for all instances of the item class. Finally it provides procedures for adding and removing event listeners to and from an item or a class.

Note If GRTL is used with G2 Version 8.0 or later, this mechanism is based on the G2 Publish/Subscribe facility. For details, see the *G2 Reference Manual*.

Classes

grtl-event-source

A class of object that supports the object model event notification. Listeners can register either at the class level or at the instance level to be notified of changes.

Class Inheritance Path

grtl-event-source, grtl-item, item

Methods

grtl-event-source::grtl-add-attribute-change-event-listener
grtl-event-source::grtl-add-event-listener
grtl-event-source::grtl-disable-event-notification
grtl-event-source::grtl-dispatch-event
grtl-event-source::grtl-enable-event-notification
grtl-event-source::grtl-is-event-notification-enabled
grtl-event-source::grtl-remove-all-event-listeners
grtl-event-source::grtl-remove-attribute-event-listener
grtl-event-source::grtl-remove-event-listener

grtl-event-source-class-definition

A class definition with additional attributes for adding a class description and extended attribute information. The event source keeps a list of registered listeners that should be notified of changes in all instances of this class.

Class Inheritance Path

grtl-event-source-class-definition, grtl-class-definition, class-definition, definition, item

Attributes

Attribute	Description
class-description	See grtl-class-definition.
Allowable values:	inherited
Default value:	""
instance-key-attribute-name	See grtl-class-definition.
Allowable values:	inherited
Default value:	UUID
class-attribute-properties	See grtl-class-definition.
Allowable values:	inherited
Default value:	sequence ()

Methods and Procedures

Methods

class-definition::grtl-add-event-listener
class-definition::grtl-remove-all-event-listeners
class-definition::grtl-remove-event-listener
grtl-datapoint::grtl-add-datapoint-change-event-listener
grtl-datapoint::grtl-remove-datapoint-change-event-listener
grtl-event-source::grtl-add-attribute-change-event-listener
grtl-event-source::grtl-add-event-listener
grtl-event-source::grtl-disable-event-notification
grtl-event-source::grtl-dispatch-event
grtl-event-source::grtl-enable-event-notification
grtl-event-source::grtl-is-event-notification-enabled
grtl-event-source::grtl-remove-all-event-listeners
grtl-event-source::grtl-remove-attribute-event-listener
grtl-event-source::grtl-remove-event-listener
item::grtl-disable-event-notification
item::grtl-dispatch-event
item::grtl-enable-event-notification
item::grtl-event-notification
item::grtl-is-event-notification-enabled

Procedures

grtl-add-remote-event-listener-to-class
grtl-add-remote-event-listener-to-item
grtl-remove-remote-event-listener-to-class
grtl-remove-remote-event-listener-to-item

class-definition::grtl-add-event-listener

Synopsis

class-definition::grtl-add-event-listener
(*event-source*: class-definition, *listener*: item-or-value, *event-type*: symbol, *user-data*: sequence)

Argument	Description
event-source	The event source class.
listener	The event listener, which can be an object or any other key. The listener is passed as the first argument in the notification procedure to support object dispatching via methods.
event-type	The event type to which to subscribe, as a symbol.
user-data	Any user data passed into the notification method.

Description

Registers a listener with an event source class. As changes occur in any instance, such as attribute changes, listeners are notified of the change. Subscribers can register as listeners within the same G2 process, within a GSI process, or within another G2 process. Also, subscribers can register in synchronous mode, whereby the notification method is "called," or in asynchronous mode, whereby the notification method is "started."

The notification system supports listeners who subscribe within the same G2 process, within a remote G2 process, within a G2 Gateway process, within a G2 JavaLink process, or within a G2 ActiveXLink process. The notification uses a fire-and-forget mode, that is, it starts the notification callback procedure and does not wait for return values or aborts within the callback.

Note that the current implementation does not support registering for attribute value change notifications (grtl-properties-changed event type) and datapoint value change notifications (grtl-datapoint-value-changed event type) at the class level.

For details, see grtl-event-source::grtl-add-event-listener.

class-definition::grtl-remove-all-event-listeners

Synopsis

class-definition::grtl-remove-all-event-listeners
(*event-source*: class-definition)

Argument	Description
event-source	The event source class.

Description

Deregisters all listeners for any instance of the specified class and subclasses.

class-definition::grtl-remove-event-listener

Synopsis

class-definition::grtl-remove-event-listener
(*event-source*: class-definition, *listener*: item-or-value, *event-type*: symbol)

Argument	Description
event-source	The event source class.
listener	The event listener, which can be an object or any other key. The listener is passed as the first argument in the notification procedure to support object dispatching via methods.
event-type	The event type from which to unsubscribe.

Description

Deregisters a listener with an event source class.

grtl-datapoint::grtl-add-datapoint-change-event-listener

Synopsis

grtl-datapoint::grtl-add-datapoint-change-event-listener
(*datapoint*: grtl-datapoint , *listener*: item-or-value , *user-data*: sequence)

Argument	Description
datapoint	The event source object that sends event notification.
listener	The event listener, which can be an object or any other key. The listener is passed as the first argument in the notification procedure to support object dispatching via methods.
event-type	The event type from which to unsubscribe.
user-data	Any user data passed into the notification method.

Description

Registers a listener with a *grtl-datapoint*. The listener is notified of value changes of the datapoint. Note that the current implementation requires that you use grtl-set-value or grtl-set-property-value to send the notification. Except for grtl-external-datapoint instances, concluding a value to a variable-or-parameter does not send notification of the event when the value changes. When the value of the datapoint changes, the listener receives the grtl-datapoint-value-changed event type. The arguments of this event type include the current value of the datapoint, the status of the datapoint (the symbol ok or bad), and the key of the grtl-datapoint.

Here is an example of the notification structure. Note that it is important for the listener to test the event type in its grtl-event-notification method.

```
structure
(EVENT-TYPE: the symbol grtl-datapoint-value-changed,
DATAPOINT-VALUE:12.2,
DATAPOINT-COLLECTION-time: 340,
DATAPOINT-KEY: "sensor-3.pv",
DATAPOINT-STATUS: the symbol ok)
```

grtl-datapoint::grtl-remove-datapoint-change-event-listener

Synopsis

grtl-datapoint::grtl-remove-datapoint-change-event-listener
(*datapoint*: grtl-datapoint, *listener*: item-or-value)

Argument	Description
datapoint	The event source object that sends event notification.
listener	The event listener, which can be an object or any other key. The listener is passed as the first argument in the notification procedure to support object dispatching via methods.

Description

Deregisters a listener with a grtl-datapoint. The listener will not be notified of value changes of the datapoint after it has been deregistered. The datapoint needs to be in a subtable of an item, otherwise the error grtl-cannot-add-listener-to-datapoint is signalled.

grtl-event-source::grtl-add-attribute-change-event-listener

Synopsis

grtl-event-source::grtl-add-attribute-change-event-listener
(*event-source*: grtl-event-source, *listener*: item-or-value,
attribute-names: sequence, *user-data*: sequence)

Argument	Description
event-source	The event source object that sends event notification.
listener	The event listener, which can be an object or any other key. The listener is passed as the first argument in the notification procedure to support object dispatching via methods.
attribute-names	A sequence of attribute names, as symbols.
user-data	Any user data passed into the notification method.

Description

Registers a listener with a event source to be notified of attribute changes. When the value of an attribute changes, the listener receives the grtl-properties-changed event. The arguments of this event type includes the list properties that changed.

Here is an example of the notification structure. Note that it is important for the listener to test the event type in its grtl-event-notification method.

```
structure
(EVENT-TYPE: the symbol grtl-properties-changed,
CHANGED-PROPERTIES: sequence (the symbol temperature, the symbol
label))
```

Note that if a listener attempts to register multiple times to the same event source to be notified of attribute changes, the list of attributes for the different calls is not merged together. Previous calls are automatically unregistered and only the last one is kept.

grtl-event-source::grtl-add-event-listener

Synopsis

grtl-event-source::grtl-add-event-listener
(*event-source*: grtl-event-source, *listener*: item-or-value, *event-type*: symbol, *user-data*: sequence)

Argument	Description
event-source	The event source class.
listener	The event listener, which can be an object or any other key. The listener is passed as the first argument in the notification procedure to support object dispatching via methods.
event-type	The event type to which to subscribe, as a symbol.
user-data	Any user data passed into the notification method.

Description

Registers a listener with an event source object. As changes occur, you can call grtl-dispatch-event to notify listeners of changes. Specific events are identified by their event type, which is a symbol. The event notification consists of a G2 structure passed to callback procedures. This structure contains at least one symbolic attribute called event-type, which specifies the type of event. The structure may contain additional attributes depending on the event type. The notification system supports notified listeners who subscribed within the same G2 process, within a remote G2 process, within a G2 Gateway process, within a G2 JavaLink process, or within a G2 ActiveXLink process. The notification uses a fire-and-forget mode, which means it starts the notification callback procedure and does not deal with return values or aborts within the callback.

The method on the listener that is called to post notification events is grtl-event-notification, which needs to be implemented for each listener class. This method should test for the different event types by checking the event-type attribute of the event structure. The signature of this method is:

grtl-event-notification
(*listener*: class item, *event-source*: class grtl-event-source, *event*: structure, *user-data*: sequence)

For remote listeners, that is, listeners in remote G2, G2 Gateway, G2 JavaLink, or G2 ActiveXLink processes, the callback procedure name is called grtl-remote-event-notification. Therefore bridges (Java, C, C++) must implement this RPC, which will be called by G2 to notify the remote processes of changes.

For applications that use G2 ActiveXLink, there is only one COM procedure (in VB, for example) that is called from G2, and one of the arguments specifies the desired procedure that is requested. In this case, this argument is grtl-remote-event-notification and the code (typically VB) will need to dispatch, based on this key as required by the application.

grtl-event-source::grtl-disable-event-notification

Synopsis

grtl-event-source::grtl-disable-event-notification
(*event-source*: grtl-event-source)
-> *state*: truth-value

Argument	Description
event-source	The event source object.

Return Value	Description
state	The previous event state.

Description

Disables event notification for an event source. This method returns the previous state of the event.

grtl-event-source::grtl-dispatch-event

Synopsis

grtl-event-source::grtl-dispatch-event
(*event-source*: grtl-event-source, *event*: structure)

Argument	Description
event-source	The event source object.
event	The content of the event.

Description

Notifies listeners of changes in the event source object. The *event* argument contains detailed information of the event as a structure. All events include at least one attribute in the structure called event-type, as a symbol. The *event* argument can include additional arguments that depend on the type of event, as well. The *event* argument is given as a structure to dispatch events without needing to worry about event allocation and deallocation. The event uses a single callback, grtl-event-notification, to dispatch all types of events, rather than different callbacks for different types of events. Using a single callback simplifies the dispatching of events across multiple systems, for example, using JMS message maps. For more information, see item::grtl-event-notification.

grtl-event-source::grtl-enable-event-notification

Synopsis

grtl-event-source::grtl-enable-event-notification
    (*event-source*: grtl-event-source)
    -> *state*: truth-value

Argument	Description
event-source	The event source object.

Return Value	Description
state	The previous event state.

Description

Enable event notification for an event source. This method returns the previous state of the event.

grtl-event-source::grtl-is-event-notification-enabled

Synopsis

grtl-event-source::grtl-is-event-notification-enabled
    (*event-source*: grtl-event-source)
    -> *enabled*: truth-value

Argument	Description
event-source	The event source object.

Return Value	Description
enabled	True if event notification is enabled for the event source; otherwise, returns false.

Description

Returns true if the event notification enabled for an event source.

grtl-event-source::grtl-remove-all-event-listeners

Synopsis

grtl-event-source::grtl-remove-all-event-listeners
(*event-source*: grtl-event-source)

Argument	Description
event-source	The event source object.

Description

Deregisters all listeners for any instance of the specified event source or subclass.

grtl-event-source::grtl-remove-attribute-event-listener

Synopsis

grtl-remove-attribute-event-listener
(*event-source*: class grtl-event-source, *listener*: item-or-value)

Argument	Description
event-source	The event source class.
listener	The event listener, which can be an object or any other key. The listener is passed as the first argument in the notification procedure to support object dispatching via methods.

Description

Deregisters a listener with an event source for all registered attributes. For more information, see grtl-event-source::grtl-add-attribute-change-event-listener.

grtl-event-source::grtl-remove-event-listener

Synopsis

grtl-event-source::grtl-remove-event-listener
(*event-source*: class grtl-event-source, *listener*: item-or-value,
event-type: symbol)

Argument	Description
event-source	The event source class.
listener	The event listener, which can be an object or any other key. The listener is passed as the first argument in the notification procedure to support object dispatching via methods.
event-type	The event type to which to unsubscribe.

Description

Deregisters a listener with an event source for a particular event type. For more information, see class-definition::grtl-add-event-listener.

item::grtl-disable-event-notification

Synopsis

item::grtl-disable-event-notification
    (*event-source*: grtl-event-source)

Argument	Description
event-source	The event source object.

Description

Disables event notification for an event source.

item::grtl-dispatch-event

Synopsis

grtl-event-source::grtl-dispatch-event
(*event-source*: grtl-event-source, *event*: structure)

Argument	Description
event-source	The event source object.
event	The content of the event.

Description

Notifies listeners of changes in the event source object. The *event* argument contains detailed information of the event as a structure. All events include at least one attribute in the structure called event-type, as a symbol. The *event* argument can include additional arguments that depend on the type of event, as well. The *event* argument is given as a structure to dispatch events without needing to worry about event allocation and deallocation. The event uses a single callback, grtl-event-notification, to dispatch all types of events, rather than different callbacks for different types of events. Using a single callback simplifies the dispatching of events across multiple systems, for example, using JMS message maps. For more information, see item::grtl-event-notification.

item::grtl-enable-event-notification

Synopsis

item::grtl-enable-event-notification
(*event-source*: item)
-> *state*: truth-value

Argument	Description
event-source	The event source object.

Return Value	Description
state	The previous event state.

Description

Enables event notification for an event source and returns the previous event state.

item::grtl-event-notification

Synopsis

item::grtl-event-notification
(*listener*: item, *event-source*: item-or-value, *event*: structure,
user-data: sequence)

Argument	Description
listener	The listener to notify of the event.
event-source	The item that initiated the event or the key of the item as a text for events received from a remote host.
event	The content of the event.
user-data	Any user data passed into the notification method.

Description

Notifies a listener of an event from an event source. Events are dispatched by calls to grtl-dispatch-event on the *event-source*, which calls grtl-event-notification for every listener.

item::grtl-is-event-notification-enabled

Synopsis

item::grtl-is-event-notification-enabled
    (*event-source*: grtl-event-source)
    -> *enabled*: truth-value

Argument	Description
event-source	The event source object.

Return Value	Description
enabled	True if event notification is enabled for the event source; otherwise, returns false.

Description

Returns true if the event notification enabled for an event source.

grtl-add-remote-event-listener-to-class

Synopsis

grtl-add-remote-event-listener-to-class
(*event-source-class-name*: symbol, *listener*: value, *sync*: truth-value, *selector*: text , *user-data*: sequence)

Argument	Description
event-source-class-name	The event source class.
listener	The event listener, which can only be a value for remote listeners.
sync	True to use synchronous notification; otherwise, false to use asynchronous notification.
selector	An SQL-like selector that describes which events should send notification to listeners.
user-data	Any user data passed into the notification method.

Description

Registers a remote listener to a class.

grtl-add-remote-event-listener-to-item

Synopsis

grtl-add-remote-event-listener-to-item
(*event-source-key*: symbol, *event-source-class-name*: symbol, *listener*: value, *sync*: truth-value, *selector*: text , *user-data*: sequence)

Argument	Description
event-source-key	The event source instance that sends notification of events.
event-source-class-name	The event source class.
listener	The event listener, which can only be a value for remote listeners.
sync	True to use synchronous notification; otherwise, false to use asynchronous notification.
selector	An SQL-like selector that describes which events should send notification to listeners.
user-data	Any user data passed into the notification method.

Description

Registers a remote listener when object passing is not possible or if the *event-source-key* is the only information known by the remote process. In distributed applications, a remote process may not be able to refer to the event source object.

grtl-remove-remote-event-listener-to-class

Synopsis

grtl-remove-remote-event-listener-to-class
(*event-source-class-name*: symbol, *listener*: value, *sync*: truth-value, *selector*: text)

Argument	Description
event-source-class-name	The event source class.
listener	The event listener, which can only be a value for remote listeners.
sync	True to use synchronous notification; otherwise, false to use asynchronous notification.
selector	An SQL-like selector that describes which events should send notification to listeners.

Description

Deregisters a remote listener with an event source class.

grtl-remove-remote-event-listener-to-item

Synopsis

grtl-remove-remote-event-listener-to-item
    (event-source-key: text, event-source-class-name: symbol, listener: value, sync: truth-value, selector: text)

Argument	Description
event-source-key	The event source instance that sends notification of events.
event-source-class-name	The event source class.
listener	The event listener, which can only be a value for remote listeners.
sync	True to use synchronous notification; otherwise, false to use asynchronous notification.
selector	An SQL-like selector that describes which events should send notification to listeners.

Description

Deregisters a remote listener with an event source object.

32

User Interface

Describes methods to support user interface operations of the object model.

Introduction

GRTL provides a workspace and indicator arrow class, along with methods on the item class for highlighting items with the arrow.

Classes

grtl-workspace
grtl-arrow

grtl-workspace

An extension of the kb-workspace class.

Class Inheritance Path

grtl-workspace, kb-workspace, item

Attributes

Attribute	Description
grid-size	Will be used in a future release to implement a workspace grid.
Allowable values:	Any integer
Default value:	0

grtl-arrow

The indicator that grtl-view-object uses to show and point to an item.

Class Inheritance Path

grtl-arrow, grtl-object, object, grtl-item, item

Attributes

Attribute	Description
note	The text that is displayed next to the indicator arrow and passed as an argument to the grtl-view-object API.
Allowable values:	Any text
Default value:	""

Methods

item::grtl-show-properties

Synopsis

item::grtl-show-properties
    (*itm*: item, *client*: ui-client-item)
    -> *applied*: truth-value

Argument	Description
itm	The object whose properties to show.
client	Display it in this view.

Return Value	Description
applied	Whether the property values were applied to the object.

Description

Displays the properties dialog of an item, if defined. This method returns true if the OK button of the dialog is pressed and changes are applied; otherwise, it returns false if no changes were applied.

This method should be implemented for specific subclasses to display the appropriate properties dialog. The user interface can then rely on this generic method.

item::grtl-view-object

Synopsis

item::grtl-view-object
    (*itm*: item, *note*: text, *client*: ui-client-item)

Argument	Description
itm	The object to view.
note	A text to display next to the indicator arrow.
client	The client in which to display the object, typically a G2 window.

Description

Displays an object with an indicator arrow. The indicator arrow is removed after 30 seconds. You can also click the arrow to remove it.

item::grtl-view-object

Synopsis

item::grtl-view-object
(*itm*: item, *note*: text, *hide-after-timeout*: truth-value, *timeout*: integer, *indicator-color*: symbol, *client*: ui-client-item)

Argument	Description
itm	The object to view.
note	A text to display next to the item indicator.
hide-after-timeout	True to hide the item indicator after a specified timeout.
timeout	The timeout in seconds after which the indicator is hidden if *hide-after-timeout* is true.
indicator-color	The color of the indicator arrow.
client	Display it in this view.

Description

Displays an object with an indicator arrow. You specify whether to hide the indicator arrow, the timeout, and color.

item::grtl-view-object

Synopsis

item::grtl-view-object
(*itm*: item, *show-workspace*: truth-value, *note*: text,
hide-after-timeout: truth-value, *timeout*: integer, *indicator-color*: symbol,
client: ui-client-item)

Argument	Description
itm	The object to view.
show-workspace	If true and *itm* is a workspace, displays the workspace on the window. If false, this method does not redisplay the workspace if it is not already visible.
note	A text to display next to the item indicator.
hide-after-timeout	True to hide the item indicator after a specified timeout.
timeout	The timeout in seconds after which the indicator is hidden if *hide-after-timeout* is true.
indicator-color	The color of the indicator arrow.
client	The client, typically a G2 window.

Description

Displays an object with an indicator arrow. You can specify text to display with the arrow, whether to hide the indicator arrow, a timeout, and the color.

33

Instance Repository

Describes the instance repository, which is a standard mechanism for organizing newly created instances without requiring the user to know where objects are stored.

Introduction

The instance repository is a standard mechanism for organizing newly created instances without requiring the user to know where the objects are stored. The repository organizes the instances by class on the subworkspace of a grtl-class-instance-holder, which is placed on the subworkspace of a grtl-repository.

GRTL provides procedures for creating repositories, storing items in a repository, getting the repository for an item, and determining if the repository has changed.

Classes

grtl-repository

A repository for storing user-defined instances. A repository organizes the instances it stores by class into instances of a grtl-class-instance-holder, which it stores on its subworkspace. Repositories can be organized into folders.

Class Inheritance Path

grtl-repository, grtl-object-with-key, object, grtl-item-with-key, grtl-item, item

Attributes

Attribute	Description
permanent-items	
Allowable values:	Any truth-value
Default value:	true
key	See grtl-item-with-key.
Allowable values:	inherited
Default value:	""

grtl-class-instance-holder

A repository for instances of a class. Instances of this class are automatically created and stored on the subworkspace of a grtl-repository to organize the instances by classes.

Class Inheritance Path

grtl-class-instance-holder, object, item

Attributes

Attribute	Description
grtl-instance-class-name	The class name of instances to store in this repository.
Allowable values:	Any symbol
Default value:	UNSPECIFIED

Procedures

grtl-find-or-create-repository

Synopsis

grtl-find-or-create-repository
(*repository-key*: text, *permanent-items*: truth-value, *module*: symbol)
-> *repository*: class grtl-repository

Argument	Description
repository-key	The key of the repository to find or create.
permanent-items	True if newly created repository should make items permanent.
module	The module name where new repositories should be stored.

Return Value	Description
repository	The resulting repository.

Description

Returns the grtl-repository used to store instances given a repository key. If necessary, this procedure creates a new repository and stores it in the bin workspace of the specified module.

grtl-find-repository

Synopsis

grtl-find-repository
    (*repository-key*: text)
    -> *repository*: class grtl-repository

Argument	Description
repository-key	The key of the repository to find.

Return Value	Description
repository	The resulting repository.

Description

Returns the grtl-repository used to store instances given a repository key. The error grtl-repository-not-found is signalled if no repository is found.

grtl-get-change-flag

Synopsis

grtl-get-change-flag
    (*repository*: class grtl-repository)
    -> *changed*: truth-value

Argument	Description
repository	The repository whose change flag to get.

Return Value	Description
changed	True if the content of the repository changed since the last reset.

Description

Gets the value of the change flag for a repository. The user interface can use the change flag to determine whether the file containing the repository needs to be saved to disk.

grtl-get-default-repository

Synopsis

grtl-get-default-repository
    ( )
    -> *repository*: class grtl-repository

Return Value	Description
repository	The default repository.

Description

Returns the default grtl-repository used to store instances.

grtl-is-item-in-repository

Synopsis

grtl-is-item-in-repository
    (*itm*: item)
    -> *in-repository*: truth-value

Argument	Description
itm	The item whose repository to check.

Return Value	Description
in-repository	True if the specified item is in a repository; otherwise, false.

Description

Returns true if an item is in a repository; otherwise, returns false.

grtl-reset-change-flag

Synopsis

grtl-reset-change-flag
    (*repository*: class grtl-repository)

Argument	Description
repository	The repository whose change flag to reset.

Description

Resets the value of the change flag for a repository. The user interface can use the change flag to determine whether the file containing the repository needs to be saved to disk.

grtl-set-change-flag

Synopsis

grtl-set-change-flag
    (*repository*: grtl-repository)

Argument	Description
repository	The repository whose change flag to set.

Description

Sets the value of the change flag for a repository. The user interface can use the change flag to determine whether the file containing the repository needs to be saved to disk.

grtl-store-new-items

Synopsis

grtl-store-new-items
    (*repository*: class grtl-repository, *new-items*: sequence)

Argument	Description
repository	The repository in which to store the items.
new-items	The new items to store.

Description

Stores one or more items within the specified repository.

grtl-store-new-items-in-default-repository

Synopsis

grtl-store-new-items-in-default-repository
    (*new-items*: sequence)
    -> *repository*: class grtl-repository

Argument	Description
new-items	The new items to store.

Return Value	Description
repository	The repository in which the items are stored.

Description

Stores one or more items in the default repository. This procedure returns the repository in which the items are stored. If necessary, this procedure creates a new repository and stores it in the bin workspace.

34

Datapoints

Describes datapoint classes and methods, including classes and methods to represent datapoints in process maps.

Introduction

GRTL provides a number of datapoint classes, which are event sources with keys. Each datapoint refers to a source datapoint that provides data to the datapoint. The datapoint keeps a history of values over time, which enables you to perform calculations on data histories and plot those values over time in a trend chart.

GRTL provides various types of datapoints:

- Generic datapoints, which provide the underlying facilities for datapoint management including activating, deactivating, and evaluating datapoints; relating and unrelating datapoints to and from their source datapoints; and getting and setting the status.
- Derived datapoints, which get their values based on a formula that can refer to the values of other datapoints.
- External datapoints, which receive values for PV, SP, OP, and Mode tags from an external DCS system.
- Simple datapoints, which are internal datapoints defined on a domain object in a process map and typically get their values from external or derived datapoints. See Process Maps, Domain Objects, and Schemes.
- Datapoint displays, which allow you to display the current value of a source datapoint in a readout table or trend chart.

In addition, GRTL provides a datapoint class definition, which inherits from grtl-class-definition and grtl-event-source-class-definition for defining your own datapoint classes.

Each type of datapoint defines subclasses for all G2 data types: float, integer, logical, quantity, symbolic, and text. Each of these specific datapoint types provides built-in logging capabilities by inheriting from the grtl-logged-item class, which logs datapoint values.

Datapoint Class and Methods

Classes

grtl-datapoint
grtl-datapoint-class-definition

Methods

grtl-datapoint::grtl-get-datapoint-status
grtl-datapoint::grtl-get-units
grtl-datapoint::grtl-initialize
grtl-datapoint::grtl-relate-datapoint-sources
grtl-datapoint::grtl-set-datapoint-status
grtl-datapoint::grtl-show-properties
grtl-datapoint::grtl-unrelate-datapoint-sources
grtl-datapoint::grtl-update-datapoint-status
variable-or-parameter::grtl-get-value
variable-or-parameter::grtl-set-value
variable-or-parameter::grtl-set-value-with-collection-time
variable-or-parameter::grtl-update-datapoint-listeners

grtl-datapoint

An abstract class that is the superior class for all types of datapoints. Datapoints inherit from grtl-item-with-key and grtl-event-source.

Class Inheritance Path

grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-item, item

Attributes

Attribute	Description
description	A text description of the datapoint.
Allowable values:	Any text
Default value:	""
units	The units in which the datapoint keeps its data.
Allowable values:	Any text
Default value:	""
status	The status of the datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	The key of the source item that provides data to the datapoint.
Allowable values:	Any value
Default value:	""
nb-of-historical-values	The maximum number of historical values to keep in the datapoint history.
Allowable values:	Any integer
Default value:	0
maximum-history-age	The maximum age of historical values to keep in the datapoint history.

Allowable values: Any item or value

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

Methods

grtl-datapoint-class-definition
grtl-datapoint::grtl-get-datapoint-status
grtl-datapoint::grtl-get-units
grtl-datapoint::grtl-initialize
grtl-datapoint::grtl-relate-datapoint-sources
grtl-datapoint::grtl-set-datapoint-status
grtl-datapoint::grtl-show-properties
grtl-datapoint::grtl-unrelate-datapoint-sources
grtl-datapoint::grtl-update-datapoint-status

grtl-datapoint-class-definition

A class definition for defining datapoint classes. This class definition inherits from grtl-class-definition and grtl-event-source-class-definition.

Class Inheritance Path

grtl-datapoint-class-definition, grtl-event-source-class-definition, grtl-class-definition, class-definition, definition, item

Attributes

Attribute	Description
class-description	See grtl-class-definition.
Allowable values:	inherited
Default value:	""
instance-key-attribute-name	See grtl-class-definition.
Allowable values:	inherited
Default value:	UUID
class-attribute-properties	See grtl-class-definition.
Allowable values:	inherited
Default value:	sequence ()

grtl-datapoint::grtl-get-datapoint-status

Synopsis

grtl-datapoint::grtl-get-datapoint-status
    (*datapoint*: grtl-datapoint)
    -> *status*: symbol

Argument	Description
datapoint	The datapoint whose status to get.

Return Value	Description
status	The status of the current datapoint value.

Description

Gets the status of the current datapoint value.

grtl-datapoint::grtl-get-units

Synopsis

grtl-datapoint::grtl-get-units
    (*datapoint*: grtl-datapoint)
    -> *units*: text

Argument	Description
datapoint	The datapoint whose units to get.

Return Value	Description
units	The units of a datapoint.

Description

Gets the units of a datapoint.

grtl-datapoint::grtl-initialize

Synopsis

grtl-datapoint::grtl-initialize
    (*datapoint*: grtl-datapoint)

Argument	Description
datapoint	The datapoint to initialize.

Description

Initializes the datapoint by subscribing to event sources.

grtl-datapoint::grtl-relate-datapoint-sources

Synopsis

grtl-datapoint::grtl-relate-datapoint-sources
    (*datapoint*: grtl-datapoint)

Argument	Description
datapoint	The datapoint whose source datapoints to relate.

Description

Relates the source datapoints to the specified datapoint.

grtl-datapoint::grtl-set-datapoint-status

Synopsis

grtl-datapoint::grtl-set-datapoint-status
    (*datapoint*: grtl-datapoint, *status*: symbol)

Argument	Description
datapoint	The datapoint whose status to set.
status	The status of the current datapoint value.

Description

Sets the status of the current datapoint value.

grtl-datapoint::grtl-show-properties

Synopsis

grtl-datapoint::grtl-show-properties
(*datapoint*: grtl-datapoint, *client*: ui-client-item)
-> *activated*: truth-value

Argument	Description
datapoint	The datapoint whose properties to show.
client	The client, typically a G2 window.

Return Value	Description
activated	Whether the property values were applied to the object.

Description

Displays the properties dialog of the specified datapoint. This method returns true if the OK button of the dialog is pressed and changes are applied; otherwise, it returns false if no changes were applied.

grtl-datapoint::grtl-unrelate-datapoint-sources

Synopsis

grtl-datapoint::grtl-unrelate-datapoint-sources
(*datapoint*: grtl-datapoint)

Argument	Description
datapoint	The datapoint to unrelate.

Description

Unrelates the source datapoints from the specified datapoint.

grtl-datapoint::grtl-update-datapoint-status

Synopsis

grtl-datapoint::grtl-update-datapoint-status
(*datapoint*: grtl-datapoint)

Argument	Description
datapoint	The datapoint whose status to update.

Description

Updates the status of the specified datapoint. The default implementation of this method does nothing. Subclasses should implement the specific behavior.

variable-or-parameter::grtl-get-value

Synopsis

variable-or-parameter::grtl-get-value
    (*item*: variable-or-parameter)
    -> *current-value*: value, *collection-time*: quantity

Argument	Description
item	The variable or parameter whose value to get.

Return Value	Description
current-value	The current value.
collection-time	The collection time.

Description

Gets the current value of a variable or parameter.

variable-or-parameter::grtl-set-value

Synopsis

variable-or-parameter::grtl-set-value
(*item*: variable-or-parameter, *new-value*: value)

Argument	Description
itm	The variable or parameter whose value to set.
new-value	The new value.

Description

Sets the current value of a variable or parameter.

variable-or-parameter::grtl-set-value-with-collection-time

Synopsis

variable-or-parameter::grtl-set-value-with-collection-time
(*item*: variable-or-parameter, *new-value*: value, *collection-time*: quantity)

Argument	Description
item	The variable or parameter whose value to set.
new-value	The new value.
collection-time	The collection time of the value.

Description

Sets the current value of a variable or parameter, including its collection time.

variable-or-parameter::grtl-update-datapoint-listeners

Synopsis

variable-or-parameter::grtl-update-datapoint-listeners
(*item*: variable-or-parameter)

Argument	Description
item	The variable or parameter that should send a notification to all listeners.

Description

Sends a notification to all listeners of the variable or parameter value.

Derived Datapoint Class and Methods

Classes

grtl-derived-datapoint
grtl-derived-float-datapoint
grtl-derived-integer-datapoint
grtl-derived-logical-datapoint
grtl-derived-quantitative-datapoint
grtl-derived-symbolic-datapoint
grtl-derived-text-datapoint

Methods

grtl-derived-datapoint::grtl-compile
grtl-derived-datapoint::grtl-evaluate
grtl-derived-datapoint::grtl-get-formula
grtl-derived-datapoint::grtl-initialize
grtl-derived-datapoint::grtl-set-formula
grtl-derived-datapoint::grtl-show-properties
grtl-derived-datapoint::grtl-update-datapoint-status

grtl-derived-datapoint

An abstract class used to calculate a formula that refers to one or more datapoints.

Class Inheritance Path

grtl-derived-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-item, item

Attributes

Attribute	Description
dp-formula	The formula to calculate as the datapoint value.
Allowable values:	Any text
Default value:	""
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited
Default value:	0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

Methods

grtl-derived-datapoint::grtl-compile
grtl-derived-datapoint::grtl-evaluate
grtl-derived-datapoint::grtl-get-formula
grtl-derived-datapoint::grtl-initialize
grtl-derived-datapoint::grtl-set-formula
grtl-derived-datapoint::grtl-show-properties
grtl-derived-datapoint::grtl-update-datapoint-status

grtl-derived-float-datapoint

A derived datapoint that stores float values. Derived datapoints also inherit from grtl-logged-item.

Class Inheritance Path

grtl-derived-float-datapoint, float-parameter, quantitative-parameter, parameter, variable-or-parameter, object, grtl-derived-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
dp-formula	See grtl-derived-datapoint.
Allowable values:	inherited
Default value:	""
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-derived-integer-datapoint

A derived datapoint that stores integer values. Derived datapoints also inherit from grtl-logged-item.

Class Inheritance Path

grtl-derived-integer-datapoint, integer-parameter, quantitative-parameter, parameter, variable-or-parameter, object, grtl-derived-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
dp-formula	See grtl-derived-datapoint.
Allowable values:	inherited
Default value:	""
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-derived-logical-datapoint

A derived datapoint that stores truth values. Derived datapoints also inherit from grtl-logged-item.

Class Inheritance Path

grtl-derived-logical-datapoint, logical-parameter, parameter, variable-or-parameter, object, grtl-derived-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
dp-formula	See grtl-derived-datapoint.
Allowable values:	inherited
Default value:	""
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-derived-quantitative-datapoint

A derived datapoint that stores quantity values. Derived datapoints also inherit from grtl-logged-item.

Class Inheritance Path

grtl-derived-quantitative-datapoint, quantitative-parameter, parameter, variable-or-parameter, object, grtl-derived-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
dp-formula	See grtl-derived-datapoint.
Allowable values:	inherited
Default value:	""
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-derived-symbolic-datapoint

A derived datapoint that stores symbolic values. Derived datapoints also inherit from grtl-logged-item.

Class Inheritance Path

grtl-derived-symbolic-datapoint, symbolic-parameter, parameter, variable-or-parameter, object, grtl-derived-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
dp-formula	See grtl-derived-datapoint.
Allowable values:	inherited
Default value:	""
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-derived-text-datapoint

A derived datapoint that stores text values. Derived datapoints also inherit from grtl-logged-item.

Class Inheritance Path

grtl-derived-text-datapoint, text-parameter, parameter, variable-or-parameter, object, grtl-derived-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
dp-formula	See grtl-derived-datapoint.
Allowable values:	inherited
Default value:	""
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-derived-datapoint::grtl-compile

Synopsis

grtl-derived-datapoint::grtl-compile
(*ddp*: grtl-derived-datapoint)
-> *derivation-procedure*: class grtl-derivation-procedure

Argument	Description
datapoint	The derived datapoint to compile.

Return Value	Description
derivation-procedure	The derivation procedure into which the formula was compiled.

Description

Compiles the formula for a derived datapoint and returns a procedure, which the caller must manage. Typically, managing the procedure involves making it permanent and ensuring the module where it is stored gets saved. This method signals the error grtl-no-formula-specified if no formula is specified, grtl-no-server-key-specified if the derived datapoint does not have a server key, and grtl-could-not-compile-formula if the compilation fails.

grtl-derived-datapoint::grtl-evaluate

Synopsis

grtl-derived-datapoint::grtl-evaluate
    (*datapoint*: grtl-derived-datapoint)

Argument	Description
datapoint	The derived datapoint to evaluate.

Description

Calls the derivation procedure for the derived datapoint, which gets values from all the servers of the derived datapoint, evaluates the formula.

grtl-derived-datapoint::grtl-get-formula

Synopsis

grtl-derived-datapoint::grtl-get-formula
    (*datapoint*: grtl-derived-datapoint)
    -> *formula*: text

Argument	Description
datapoint	The derived datapoint whose formula to get.

Return Value	Description
formula	The formula of the derived datapoint.

Description

Returns the formula of a derived datapoint.

grtl-derived-datapoint::grtl-initialize

Synopsis

grtl-derived-datapoint::grtl-initialize
(*datapoint*: grtl-derived-datapoint)

Argument	Description
datapoint	The datapoint to initialize.

Description

Initializes the datapoint by subscribing to event sources.

grtl-derived-datapoint::grtl-set-formula

Synopsis

grtl-derived-datapoint::grtl-set-formula
(*datapoint*: grtl-derived-datapoint, *formula*: text, *compile*: truth-value)
-> *formula*: item-or-value

Argument	Description
datapoint	The derived datapoint whose formula to set.
formula	The new formula for the derived datapoint.
compile	True to compile the formula; otherwise, false.

Return Value	Description
formula	A transient procedure that is the compiled formula, which you must manage, or the symbol no-compile.

Description

Sets and optionally compiles a new formula for a derived datapoint. The formula must be a text containing any one or more datapoint keys, numerical constants, mathematical operators, and G2 functions. Each key must be bounded by vertical bars, for example, |FC101.HH|. The syntax, operators, and functions that you can use include any G2 function, including user-defined functions. If a compilation is specified, the return argument is a transient procedure, which the caller must manage. Typically, this means making the procedure permanent and ensuring the module where the procedure is stored gets saved. This method signals the error grtl-could-not-compile-formula if the compilation fails.

grtl-derived-datapoint::grtl-show-properties

Synopsis

grtl-derived-datapoint::grtl-show-properties
(*datapoint*: grtl-derived-datapoint, *client*: ui-client-item)
-> *activated*: truth-value

Argument	Description
datapoint	The object to view.
client	Display it in this view.

Return Value	Description
activated	Whether the property values were applied to the object.

Description

Displays the properties dialog of the specified datapoint. This method returns true if the OK button of the dialog is pressed and changes are applied; otherwise, it returns false if no changes were applied.

grtl-derived-datapoint::grtl-update-datapoint-status

Synopsis

grtl-derived-datapoint::grtl-update-datapoint-status
(*datapoint*: grtl-derived-datapoint)

Argument	Description
datapoint	The datapoint whose status to update.

Description

Updates the status of the specified datapoint. The default implementation of this method does nothing. Subclasses should implement the specific behavior.

External Datapoint Class and Methods

Classes

grtl-external-datapoint
grtl-external-float-datapoint
grtl-external-integer-datapoint
grtl-external-logical-datapoint
grtl-external-quantitative-datapoint
grtl-external-symbolic-datapoint
grtl-external-text-datapoint

Methods

grtl-external-datapoint::grtl-update-datapoint-status

grtl-external-datapoint

An abstract class that is the entry point for values received from a DCS. The DCS tag can be PV (process value), SP (setpoint), OP (controller output), and mode.

Class Inheritance Path

grtl-external-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-item, item

Attributes

Attribute	Description
tag-mode	The tag type in the DCS system.
Allowable values:	PV, SP, OP, MODE
Default value:	PV
event-driven-value-propagation	Whether to propagate DCS values when the values change.
Allowable values:	Any truth-value
Default value:	true
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited

Default value: ""

nb-of-historical-values See grtl-datapoint.

Allowable values: inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

Methods

grtl-external-datapoint::grtl-update-datapoint-status

grtl-external-float-datapoint

An external datapoint that stores float values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-external-float-datapoint, float-variable, quantitative-variable, g2-variable, variable, variable-or-parameter, object, grtl-external-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
tag-mode	See grtl-external-datapoint.
Allowable values:	PV, SP, OP, MODE
Default value:	PV
event-driven-value-propagation	See grtl-external-datapoint.
Allowable values:	inherited
Default value:	true
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.

Allowable values: inherited

Default value: ""

nb-of-historical-values See grtl-datapoint.

Allowable values: inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-external-integer-datapoint

An external datapoint that stores integer values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-external-integer-datapoint, integer-variable, quantitative-variable, g2-variable, variable, variable-or-parameter, object, grtl-external-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
tag-mode	See grtl-external-datapoint.
Allowable values:	PV, SP, OP, MODE
Default value:	PV
event-driven-value-propagation	See grtl-external-datapoint.
Allowable values:	inherited
Default value:	true
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.

Allowable values: inherited

Default value: ""

nb-of-historical-values See grtl-datapoint.

Allowable values: inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-external-logical-datapoint

An external datapoint that stores truth values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-external-logical-datapoint, logical-variable, g2-variable, variable, variable-or-parameter, object, grtl-external-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
tag-mode	See grtl-external-datapoint.
Allowable values:	PV, SP, OP, MODE
Default value:	PV
event-driven-value-propagation	See grtl-external-datapoint.
Allowable values:	inherited
Default value:	true
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.

Allowable values: inherited

Default value: ""

nb-of-historical-values See grtl-datapoint.

Allowable values: inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-external-quantitative-datapoint

An external datapoint that stores quantity values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-external-quantitative-datapoint, quantitative-variable, g2-variable, variable, variable-or-parameter, object, grtl-external-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
tag-mode	See grtl-external-datapoint.
Allowable values:	PV, SP, OP, MODE
Default value:	PV
event-driven-value-propagation	See grtl-external-datapoint.
Allowable values:	inherited
Default value:	true
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.

Allowable values: inherited

Default value: ""

nb-of-historical-values See grtl-datapoint.

Allowable values: inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-external-symbolic-datapoint

An external datapoint that stores symbolic values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-external-symbolic-datapoint, symbolic-variable, g2-variable, variable, variable-or-parameter, object, grtl-external-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
tag-mode	See grtl-external-datapoint.
Allowable values:	PV, SP, OP, MODE
Default value:	PV
event-driven-value-propagation	See grtl-external-datapoint.
Allowable values:	inherited
Default value:	true
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.

Allowable values: inherited

Default value: ""

nb-of-historical-values See grtl-datapoint.

Allowable values: inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-external-text-datapoint

An external datapoint that stores text values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-external-text-datapoint, text-variable, g2-variable, variable, variable-or-parameter, object, grtl-external-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
tag-mode	See grtl-external-datapoint.
Allowable values:	PV, SP, OP, MODE
Default value:	PV
event-driven-value-propagation	See grtl-external-datapoint.
Allowable values:	inherited
Default value:	true
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.

Allowable values: inherited

Default value: ""

nb-of-historical-values See grtl-datapoint.

Allowable values: inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-external-datapoint::grtl-update-datapoint-status

Synopsis

grtl-external-datapoint::grtl-update-datapoint-status
(*datapoint*: grtl-external-datapoint)

Argument	Description
datapoint	The datapoint whose status to update.

Description

Updates the datapoint status. This method checks if the status information received from the bridge and stored in the attribute gsi-variable-status is different from 0. If so, it sets the status to bad and posts a warning message to the operator console, which is the Message Board or the default GEVM Message Browser). The status of the external datapoint is propagated to internal datapoints.

Simple Datapoint Classes

grtl-simple-datapoint

An abstract class used to represent internal datapoints, which are datapoints defined for domain objects in a process map.

Class Inheritance Path

grtl-simple-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-item, item

Attributes

Attribute	Description
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited
Default value:	0
maximum-history-age	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

grtl-simple-float-datapoint

An internal datapoint that stores float values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-simple-float-datapoint, float-parameter, quantitative-parameter, parameter, variable-or-parameter, object, grtl-simple-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited
Default value:	0
maximum-history-age	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-simple-float-datapoint-with-statistics

An internal datapoint that stores float values and calculates statistics. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-simple-float-datapoint-with-statistics, grtl-simple-float-datapoint, float-parameter, quantitative-parameter, parameter, variable-or-parameter, grtl-metrics, grtl-object, object, grtl-simple-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
count-of-statistic-periods	See grtl-metrics
Allowable values:	inherited
Default value:	0
period-start-time	See grtl-metrics
Allowable values:	inherited
Default value:	0.0
period-initial-value	See grtl-metrics
Allowable values:	inherited
Default value:	0.0
update-time	See grtl-metrics
Allowable values:	inherited
Default value:	0.0
last-change	See grtl-metrics
Allowable values:	inherited
Default value:	0.0
count-of-changes	See grtl-metrics

Allowable values: inherited

Default value: 0

sum-of-changes See grtl-metrics

Allowable values: inherited

Default value: 0.0

sum-of-incremental-changes See grtl-metrics

Allowable values: inherited

Default value: 0.0

sum-of-decremental-changes See grtl-metrics

Allowable values: inherited

Default value: 0.0

current-value See grtl-metrics

Allowable values: inherited

Default value: 0.0

min-value See grtl-metrics

Allowable values: inherited

Default value: 0.0

max-value See grtl-metrics

Allowable values: inherited

Default value: 0.0

sum-of-values See grtl-metrics

Allowable values: inherited

Default value: 0.0

average-value See grtl-metrics

Allowable values: inherited

Default value: 0.0

moving-average See grtl-metrics

Allowable values: inherited

Default value: 0.0

moving-average-standard-deviation See grtl-metrics

Allowable values: inherited

Default value: 0.0

total-time-weighted-value See grtl-metrics

Allowable values: inherited

Default value: 0.0

time-weighted-value See grtl-metrics

Allowable values: inherited

Default value: 0.0

description See grtl-datapoint.

Allowable values: inherited

Default value: ""

units See grtl-datapoint.

Allowable values: inherited

Default value: ""

status See grtl-datapoint.

Allowable values: NO-VALUE, OK, BAD, RECONSTRUCTED

Default value: NO-VALUE

source-datapoint-keys See grtl-datapoint.

Allowable values: inherited

Default value: ""

nb-of-historical-values See grtl-datapoint.

Allowable values: inherited

Default value: 0

maximum-history-age See grtl-datapoint.

Allowable values: inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-simple-integer-datapoint

An internal datapoint that stores integer values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-simple-integer-datapoint, integer-parameter, quantitative-parameter, parameter, variable-or-parameter, object, grtl-simple-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited
Default value:	0
maximum-history-age	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-simple-logical-datapoint

An internal datapoint that stores truth values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-simple-logical-datapoint, logical-parameter, parameter, variable-or-parameter, object, grtl-simple-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited
Default value:	0
maximum-history-age	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-simple-quantitative-datapoint

An internal datapoint that stores quantity values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-simple-quantitative-datapoint, quantitative-parameter, parameter, variable-or-parameter, object, grtl-simple-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited
Default value:	0
maximum-history-age	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-simple-symbolic-datapoint

An internal datapoint that stores symbolic values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-simple-symbolic-datapoint, symbolic-parameter, parameter, variable-or-parameter, object, grtl-simple-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited
Default value:	0
maximum-history-age	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

grtl-simple-text-datapoint

An internal datapoint that stores text values. External datapoints inherit from grtl-logged-item.

Class Inheritance Path

grtl-simple-text-datapoint, text-parameter, parameter, variable-or-parameter, object, grtl-simple-datapoint, grtl-datapoint, grtl-item-with-key, grtl-event-source, grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
description	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
units	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
status	See grtl-datapoint.
Allowable values:	NO-VALUE, OK, BAD, RECONSTRUCTED
Default value:	NO-VALUE
source-datapoint-keys	See grtl-datapoint.
Allowable values:	inherited
Default value:	""
nb-of-historical-values	See grtl-datapoint.
Allowable values:	inherited
Default value:	0
maximum-history-age	See grtl-datapoint.
Allowable values:	inherited

Default value: 0

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

data-logging-specification See grtl-logged-item.

Allowable values: inherited

Default value: grtl-logging-specification

Datapoint Display Classes and Methods

Classes

grtl-ui-datapoint
grtl-ui-display
grtl-process-value-display
grtl-ui-trend-chart

Methods

grtl-ui-datapoint::grtl-initialize
grtl-ui-datapoint::grtl-relate-datapoint-sources
grtl-ui-datapoint::grtl-show-properties
grtl-ui-datapoint::grtl-unrelate-datapoint-sources
grtl-ui-display::grtl-update-ui
grtl-process-value-display::grtl-update-ui
grtl-ui-trend-chart::grtl-relate-datapoint-sources
grtl-ui-trend-chart::grtl-unrelate-datapoint-sources
grtl-ui-trend-chart::grtl-update-ui

grtl-ui-datapoint

An abstract class used to represent datapoints whose values are displayed in the user interface, for example, in a process map.

Class Inheritance Path

grtl-ui-datapoint, grtl-item, item

Attributes

Attribute	Description
description	A description of the datapoint.
Allowable values:	Any text
Default value:	""
source-datapoint-key	The key of the source datapoint whose data to display.
Allowable values:	Any text
Default value:	""

Methods

grtl-ui-datapoint::grtl-initialize
grtl-ui-datapoint::grtl-relate-datapoint-sources
grtl-ui-datapoint::grtl-show-properties
grtl-ui-datapoint::grtl-unrelate-datapoint-sources

grtl-ui-display

A datapoint display that can update source datapoint values. A grtl-ui-display displays the value and the units in one line. The color of the text is changed based, on the status of the datapoint, as follows:

- ok: forest-green
- bad: red
- reconstructed: orange
- otherwise: magenta

Class Inheritance Path

grtl-ui-display, object, grtl-ui-datapoint, grtl-item, item

Attributes

Attribute	Description
description	See grtl-ui-datapoint.
Allowable values:	inherited
Default value:	""
source-datapoint-key	See grtl-ui-datapoint.
Allowable values:	inherited
Default value:	""

Methods

grtl-ui-display::grtl-update-ui

grtl-process-value-display

A datapoint display that can update source datapoint values. A grtl-display-process-value displays the value and units on the top row with the key of the datapoint below. The color of the text displaying the value with the units changes, based on the status of the datapoint, as follows:

- ok: green
- bad: red
- reconstructed: orange
- otherwise: magenta

Class Inheritance Path

grtl-process-value-display, object, grtl-ui-datapoint, grtl-item, item

Attributes

Attribute	Description
description	See grtl-ui-datapoint.
Allowable values:	inherited
Default value:	""
source-datapoint-key	See grtl-ui-datapoint.
Allowable values:	inherited
Default value:	""

Methods

grtl-process-value-display::grtl-update-ui

grtl-ui-trend-chart

A trend chart that displays datapoint values.

Class Inheritance Path

grtl-ui-trend-chart, trend-chart, grtl-ui-datapoint, grtl-item, item

Attributes

Attribute	Description
description	See grtl-ui-datapoint.
Allowable values:	inherited
Default value:	""
source-datapoint-key	See grtl-ui-datapoint.
Allowable values:	inherited
Default value:	""

Methods

grtl-ui-trend-chart::grtl-relate-datapoint-sources
grtl-ui-trend-chart::grtl-unrelate-datapoint-sources
grtl-ui-trend-chart::grtl-update-ui

grtl-ui-datapoint::grtl-initialize

Synopsis

grtl-ui-datapoint::grtl-initialize
    (*datapoint*: class grtl-ui-datapoint)

Argument	Description
datapoint	The datapoint display to initialize.

Description

Initializes the datapoint by subscribing to event sources.

grtl-ui-datapoint::grtl-relate-datapoint-sources

Synopsis

grtl-ui-datapoint::grtl-relate-datapoint-sources
(*datapoint*: class grtl-ui-datapoint)

Argument	Description
datapoint	The datapoint display whose datapoint sources to relate.

Description

Relates the source datapoints to the specified datapoint.

grtl-ui-datapoint::grtl-show-properties

Synopsis

grtl-ui-datapoint::grtl-show-properties
    (*datapoint*: grtl-ui-datapoint, *client*: ui-client-item)
    -> *activated*: truth-value

Argument	Description
datapoint	The datapoint whose properties to display.
client	The client, typically a G2 window.

Return Value	Description
activated	Whether the property values were applied to the object.

Description

Displays the properties dialog of the specified datapoint. This method returns true if the OK button of the dialog is pressed and changes are applied; otherwise, it returns false if no changes were applied.

grtl-ui-datapoint::grtl-unrelate-datapoint-sources

Synopsis

grtl-ui-datapoint::grtl-unrelate-datapoint-sources
(*datapoint*: class grtl-ui-datapoint)

Argument	Description
datapoint	The datapoint whose datapoint sources to unrelate.

Description

Unrelates the source datapoints from the specified datapoint.

grtl-ui-display::grtl-update-ui

Synopsis

grtl-ui-display::grtl-update-ui
(*viewer*: grtl-ui-display, *new-value*: value, *status*: symbol, *units*: text)

Argument	Description
viewer	The process value display to update.
new-value	The new value for the display.
status	The status of the value: no-value, ok, bad, or reconstructed.
units	The display units of the datapoint value.

Description

Refreshes the user interface display with a new value. This method is called when a new value notification event is received from a data source. It updates the label with the current value and changes the color of the display, based on the status of the datapoint: green if ok, red if bad, orange if reconstructed, and magenta otherwise.

grtl-process-value-display::grtl-update-ui

Synopsis

grtl-process-value-display::grtl-update-ui
(*viewer*: grtl-process-value-display, *new-value*: value, *status*: symbol, *units*: text)

Argument	Description
viewer	The process value display to update.
new-value	The new value for the display.
status	The status of the value: no-value, ok, bad, or reconstructed.
units	The display units of the datapoint value.

Description

Refreshes the user interface process value display with a new value. This method is called when a new value notification event is received from a data source. It updates the label with the current value and changes the color of the display, based on the status of the datapoint: green if ok, red if bad, orange if reconstructed, and magenta otherwise.

grtl-ui-trend-chart::grtl-relate-datapoint-sources

Synopsis

grtl-ui-trend-chart::grtl-relate-datapoint-sources
    (*chart*: class grtl-ui-trend-chart)

Argument	Description
chart	The trend chart whose source datapoints to relate.

Description

Relates the source datapoints to the specified trend chart.

grtl-ui-trend-chart::grtl-unrelate-datapoint-sources

Synopsis

grtl-ui-trend-chart::grtl-unrelate-datapoint-sources
(*chart*: class grtl-ui-trend-chart)

Argument	Description
chart	The trend chart whose datapoint sources to unrelate.

Description

Unrelates the source datapoints from the specified trend chart.

grtl-ui-trend-chart::grtl-update-ui

Synopsis

grtl-ui-trend-chart::grtl-update-ui
(*viewer*: grtl-ui-trend-chart, *new-value*: value, *status*: symbol, *units*: text)

Argument	Description
viewer	The process value display to update.
new-value	The new value for the display.
status	The status of the value: no-value, ok, bad, or reconstructed.
units	The display units of the datapoint value.

Description

Refreshes the user interface display with a new value. This method is called when a new value notification event is received from a data source. It adds a new datapoint to the trend chart.

Logging Classes

grtl-logged-item

A class for specifying logging for datapoints. The GDPM module implements the logging functionality that relies on instances of this class.

Class Inheritance Path

grtl-logged-item, grtl-item, item

Attributes

Attribute	Description
data-logging-specification	A logging specification.
Allowable values:	Any grtl-logging-specification
Default value:	grtl-logging-specification

grtl-logging-specification

A specification for logging datapoints. The specification determines whether logging is enabled, and can include a deadband, heartbeat interval, and minimum repeat interval.

Class Inheritance Path

grtl-logging-specification, grtl-object, object, grtl-item, item

Attributes

Attribute	Description
logging-enabled	Whether logging is enabled. By default, logging is disabled.
Allowable values:	Any truth-value
Default value:	false
deadband	The logged item logs a new value if that value differs from the last logged value by more than the deadband.
Allowable values:	Any float
Default value:	0.0
deadband-enabled	Whether the deadband is enabled.
Allowable values:	Any truth-value
Default value:	false
heartbeat-interval-in-minutes	The logged item logs values according to a schedule, rather than each time the value changes.
Allowable values:	Any float
Default value:	0.0
heartbeat-interval-enabled	Whether the heartbeat interval is enabled.
Allowable values:	Any truth-value
Default value:	false

minimum-repeat-interval-in-seconds

The logged item logs values each time it changes or once every repeat interval, whichever is longer. The data is not logged if the elapsed time from the last logged value is less than the repeat interval. You configure this attribute when the rate of incoming data is greater than the rate at which you want to log data.

Allowable values: Any integer

Default value: 0

minimum-repeat-interval-enabled

Whether the minimum repeat interval is enabled.

Allowable values: Any truth-value

Default value: false

data-logging-configuration-name

A name for the data logging specification.

Allowable values: Any symbol

Default value: NONE

35

Process Maps, Domain Objects, and Schemes

Describes definitions and operations for process maps, domain objects, and schemes.

Introduction

GRTL provides an object model for:

- Process maps, which provide hierarchical containers in which to store domain objects. Process maps inherit from grtl-repository and are event sources with keys.
- Domain objects, which are graphical representations of process equipment and define internal datapoints that obtain their data from external or derived datapoints. Domain objects can be connected to other domain objects, and stored at various levels in the process map hierarchy. This means you can implement logic that reasons over connectedness and containment relationships, for example, for event detection and diagnostic reasoning. Domain objects are event sources with keys.

- Schemes, which implement any logic used to validate datapoints, monitor situations, detect events, perform diagnostics, or perform actions. You can implement schemes as G2 procedures or using a graphical language, such as G2 Event and Data Processing (GEDP) diagrams. Datapoints can trigger one or multiple schemes in three ways: enable a scheme, disable a scheme, or evaluate a scheme.

GRTL provides various types of process map containers and a default constructor. It provides a domain object class and class definition, and corresponding methods and procedures for initializing domain objects, assigning schemes to domain objects, and getting the assigned scheme for a domain object. In addition, it provides a scheme class, a generic scheme template class, and a scheme class definition, along with methods for creating schemes from templates, and enabling, disabling, and evaluating schemes.

Process Map Classes and Methods

Classes

grtl-domain-map-decomposition
grtl-domain-map-repository
grtl-process-map-view
grtl-process-unit
grtl-production-area
grtl-production-unit

Methods

grtl-domain-map-repository::grtl-show-properties

grtl-domain-map-decomposition

A domain map that can contain other domain maps in its containment hierarchy. You can also configure the detail domain map explicitly.

Class Inheritance Path

grtl-domain-map-decomposition, grtl-domain-map-repository, grtl-repository, grtl-object-with-key, grtl-domain-object-with-key, grtl-domain-object, grtl-object, object, grtl-event-source, grtl-item-with-key, grtl-item, item

Attributes

Attribute	Description
detail-process-map-key	The key of the detail process map when not using the containment hierarchy.
Allowable values:	Any text
Default value:	""
superior-view-key	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	""
message-text-color	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	BLACK
message-background-color	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	SMOKE
permanent-items	See grtl-repository.
Allowable values:	inherited
Default value:	true
generate-cdg-events	See grtl-domain-object.

Allowable values: inherited

Default value: true

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

Methods

grtl-domain-map-decomposition::grtl-show-properties

grtl-domain-map-repository

A domain map on which you place domain objects. A domain map can be in the containment hierarchy of a superior domain map. You can also configure the superior domain map explicitly. The domain map repository can use consistent colors for messages on domain objects in the domain map.

Class Inheritance Path

grtl-domain-map-repository, grtl-repository, grtl-object-with-key, object, grtl-item-with-key, grtl-item, item

Attributes

Attribute	Description
superior-view-key	The superior domain map when not using the containment hierarchy.
Allowable values:	Any text
Default value:	""
message-text-color	The text color for all messages that occur on domain objects in the domain map.
Allowable values:	Any symbol
Default value:	BLACK
message-background-color	The background color for all messages that occur on domain objects in the domain map.
Allowable values:	Any symbol
Default value:	SMOKE
permanent-items	See grtl-repository.
Allowable values:	inherited
Default value:	true
key	See grtl-item-with-key.
Allowable values:	inherited
Default value:	""

Methods

grtl-domain-map-repository::grtl-show-properties

grtl-process-map-view

A process map on which to place domain objects. The process map can autoscale to fit a percentage of the overall screen.

Class Inheritance Path

grtl-process-map-view, grtl-workspace, kb-workspace, item

Attributes

Attribute	Description
auto-scale-to-full-screen	Whether to scale the process map to fit a percentage of the screen size when displaying it.
Allowable values:	Any truth-value
Default value:	false
auto-scale-maximum-screen-usage-percentage	The maximum percentage of the screen to use when scaling to full screen.
Allowable values:	Any float
Default value:	0.95
grid-size	See grtl-workspace.
Allowable values:	inherited
Default value:	0

grtl-process-unit

A domain map that can contain domain maps in its containment hierarchy.

Class Inheritance Path

grtl-process-unit, grtl-domain-map-decomposition, grtl-domain-map-repository, grtl-repository, grtl-object-with-key, grtl-domain-object-with-key, grtl-domain-object, grtl-object, object, grtl-event-source, grtl-item-with-key, grtl-item, item

Attributes

Attribute	Description
detail-process-map-key	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	""
superior-view-key	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	""
message-text-color	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	BLACK
message-background-color	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	SMOKE
permanent-items	See grtl-repository.
Allowable values:	inherited
Default value:	true
generate-cdg-events	See grtl-domain-object.

Allowable values: inherited

Default value: true

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

grtl-production-area

A domain map that can contain domain maps in its containment hierarchy.

Class Inheritance Path

grtl-production-area, grtl-domain-map-decomposition, grtl-domain-map-repository, grtl-repository, grtl-object-with-key, grtl-domain-object-with-key, grtl-domain-object, grtl-object, object, grtl-event-source, grtl-item-with-key, grtl-item, item

Attributes

Attribute	Description
detail-process-map-key	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	""
superior-view-key	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	""
message-text-color	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	BLACK
message-background-color	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	SMOKE
permanent-items	See grtl-repository.
Allowable values:	inherited
Default value:	true
generate-cdg-events	See grtl-domain-object.

Allowable values: inherited

Default value: true

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

grtl-production-unit

A domain map that can contain domain maps in its containment hierarchy.

Class Inheritance Path

grtl-production-unit, grtl-domain-map-decomposition, grtl-domain-map-repository, grtl-repository, grtl-object-with-key, grtl-domain-object-with-key, grtl-domain-object, grtl-object, object, grtl-event-source, grtl-item-with-key, grtl-item, item

Attributes

Attribute	Description
detail-process-map-key	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	""
superior-view-key	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	""
message-text-color	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	BLACK
message-background-color	See grtl-domain-map-repository.
Allowable values:	inherited
Default value:	SMOKE
permanent-items	See grtl-repository.
Allowable values:	inherited
Default value:	true
generate-cdg-events	See grtl-domain-object.

Allowable values: inherited

Default value: true

key See grtl-item-with-key.

Allowable values: inherited

Default value: ""

grtl-domain-map-decomposition::grtl-show-properties

Synopsis

grtl-domain-map-decomposition::grtl-show-properties
(*domain-map*: grtl-domain-map-decomposition, *client*: ui-client-item)
-> *activated*: truth-value

Argument	Description
domain-map	The domain map decomposition whose properties to show.
client	The client, typically a G2 window.

Return Value	Description
activated	Whether the property values were applied to the object.

Description

Displays the properties dialog of the specified domain map. This method returns true if the OK button of the dialog is pressed and changes are applied; otherwise, it returns false if no changes were applied.

grtl-domain-map-repository::grtl-show-properties

Synopsis

grtl-domain-map-repository::grtl-show-properties
(*domain-map*: grtl-domain-map-repository, *client*: ui-client-item)
-> *activated*: truth-value

Argument	Description
domain-map	The domain map whose properties to show.
client	The client, typically a G2 window.

Return Value	Description
activated	Whether the property values were applied to the object.

Description

Displays the properties dialog of the specified domain map. This method returns true if the OK button of the dialog is pressed and changes are applied; otherwise, it returns false if no changes were applied. See item::grtl-show-properties.

Domain Object Classes and Operations

Classes

grtl-domain-object
grtl-domain-object-class-definition
grtl-domain-object-with-key

Methods

grtl-domain-object::grtl-assign-template-schemes
grtl-domain-object::grtl-initialize-domain-item
grtl-domain-object::grtl-unassign-template-schemes
grtl-domain-object::grtl-uninitialize-domain-item
item::grtl-evaluate-scheme
item::grtl-get-scheme-by-id-for-source-object
item::grtl-get-scheme-by-template-id-for-source-object
item::grtl-get-schemes-for-source-object
kb-workspace::grtl-initialize-domain-items
kb-workspace::grtl-uninitialize-domain-items

Procedures

grtl-assign-template-schemes
grtl-get-scheme-by-id-for-source-object
grtl-get-scheme-by-template-id-for-source-object
grtl-get-schemes-for-source-object
grtl-initialize-domain-item
grtl-initialize-domain-items
grtl-is-template-for-object
grtl-unassign-template-schemes
grtl-uninitialize-domain-item
grtl-uninitialize-domain-items

Functions

grtl-is-domain-object-available

grtl-domain-object

The superior class for all domain objects. A domain object is an event source.

Class Inheritance Path

grtl-domain-object, grtl-object, object, grtl-event-source, grtl-item, item

Attributes

Attribute	Description
generate-cdg-events	Whether to enable the send-events menu choice for sending SymCure (CDG) events to the domain object.
Allowable values:	truth-value
Default value:	true
system-information-or-best-practice-url	A text string that describes system information or a URL associated with the domain object.
Allowable values:	text
Default value:	""
availability-status	Specifies the status and availability of a domain object, which is used to remove or filter out all messages if the device becomes inactive or in-repair.
Allowable values:	one of the following symbols: active, inactive, in-repair, or failed
Default value:	active

Methods

grtl-domain-object::grtl-assign-template-schemes
grtl-domain-object::grtl-initialize-domain-item
grtl-domain-object::grtl-unassign-template-schemes
grtl-domain-object::grtl-uninitialize-domain-item

grtl-domain-object-class-definition

A class definition for defining domain object classes. User-defined domain objects can be associated with a palette and a palette group.

Class Inheritance Path

grtl-domain-object-class-definition, grtl-event-source-class-definition, grtl-class-definition, class-definition, definition, item

Attributes

Attribute	Description
palette-name	The palette associated with the domain object class.
Allowable values:	Any text
Default value:	""
palette-group-name	The palette group associated with the domain object class.
Allowable values:	Any text
Default value:	""
class-description	See grtl-class-definition.
Allowable values:	inherited
Default value:	""
instance-key-attribute-name	See grtl-class-definition.
Allowable values:	inherited
Default value:	UUID
class-attribute-properties	See grtl-class-definition.
Allowable values:	inherited
Default value:	sequence ()

grtl-domain-object-with-key

A domain object with a key.

Class Inheritance Path

grtl-domain-object-with-key, grtl-domain-object, grtl-object, object, grtl-event-source, grtl-item-with-key, grtl-item, item

Attributes

Attribute	Description
generate-cdg-events	See grtl-domain-object.
Allowable values:	inherited
Default value:	true
key	See grtl-item-with-key.
Allowable values:	inherited
Default value:	""

grtl-domain-object::grtl-assign-template-schemes

Synopsis

grtl-domain-object::grtl-assign-template-schemes
(*domain-object*: grtl-domain-object, *client*: ui-client-item)

Argument	Description
domain-object	A domain object to configure.
client	The client, typically a G2 window.

Description

Assigns generic scheme templates to a domain object.

grtl-domain-object::grtl-initialize-domain-item

Synopsis

grtl-domain-object::grtl-initialize-domain-item
(*domain-object*: grtl-domain-object, *assign-templates*: truth-value,
client: ui-client-item)

Argument	Description
domain-object	A domain object to configure.
assign-templates	True to assign generic scheme templates, false otherwise.
client	The client, typically a G2 window.

Description

Initializes a domain object by compiling every derived datapoint that is an attribute of the domain object, relating every datapoint that is an attribute of the domain object to its source datapoints, if specified, and optionally assigning generic scheme templates to the domain object.

grtl-domain-object::grtl-unassign-template-schemes

Synopsis

grtl-domain-object::grtl-unassign-template-schemes
(*domain-object*: grtl-domain-object, *client*: ui-client-item)

Argument	Description
domain-object	A domain object whose scheme template to unassign.
client	The client, typically a G2 window.

Description

Unassigns a generic scheme template from a domain object.

grtl-domain-object::grtl-uninitialize-domain-item

Synopsis

grtl-domain-object::grtl-uninitialize-domain-item
(*domain-object*: grtl-domain-object, *unassign-templates*: truth-value, *client*: ui-client-item)

Argument	Description
domain-object	A domain object whose scheme template to unassign.
unassign-templates	True to unassign generic scheme templates, false otherwise.
client	The client, typically a G2 window.

Description

Uninitializes a domain object by unrelating every datapoint that is an attribute of the domain object from its source datapoints, if specified, and optionally unassigning generic scheme templates from the domain object.

item::grtl-evaluate-scheme

Synopsis

item::grtl-evaluate-scheme
    (*datapoint*: item)

Argument	Description
datapoint	The datapoint whose associated scheme should be evaluated.

Description

Evaluates a scheme associated with an item, typically a datapoint.

item::grtl-get-scheme-by-id-for-source-object

Synopsis

item::grtl-get-scheme-by-id-for-source-object
    (*source-object*: item, *key*: text)
    -> *scheme*: item-or-value

Argument	Description
source-object	The source object whose scheme to get, typically a domain object.
key	The key of the scheme to get.

Return Value	Description
scheme	The scheme associated with the specified item or false.

Description

Returns a scheme associated with a domain object given a scheme key. The key is the scheme key or scheme attribute of the scheme template. If no scheme is found, this API returns false.

item::grtl-get-scheme-by-template-id-for-source-object

Synopsis

item::grtl-get-scheme-by-template-id-for-source-object
(*source-object*: item, *key*: text)
-> *scheme*: item-or-value

Argument	Description
source-object	The source object whose scheme to get, typically a domain object.
key	The key of the scheme to get.

Return Value	Description
scheme	The scheme associated with the specified item or false.

Description

Returns a scheme associated with a domain object given a scheme key. The key is the scheme key of the scheme template. If no scheme is found, this API returns false.

item::grtl-get-schemes-for-source-object

Synopsis

item::grtl-get-schemes-for-source-object
    (*source-object*: item)
    -> *schemes*: sequence

Argument	Description
source-object	The source object whose schemes to get, typically a domain object.

Return Value	Description
schemes	A sequence of schemes for the source object.

Description

Returns a sequence of schemes for a source domain object. The sequence contains references to the schemes.

kb-workspace::grtl-initialize-domain-items

Synopsis

kb-workspace::grtl-initialize-domain-items
(*wksp*: kb-workspace, *assign-templates*: truth-value, *client*: ui-client-item)

Argument	Description
wksp	A workspace whose domain objects to initialize.
assign-templates	True to assign generic scheme templates, false otherwise.
client	The client, typically a G2 window.

Description

Configures all domain objects on a workspace by compiling every derived datapoint that is an attribute of the domain object, relating every datapoint that is an attribute of the domain object to its source datapoints, if specified, and optionally assigning generic scheme templates to the domain object.

kb-workspace::grtl-uninitialize-domain-items

Synopsis

kb-workspace::grtl-uninitialize-domain-items
(*wks*: kb-workspace, *unassign-templates*: truth-value, *client*: ui-client-item)

Argument	Description
wksp	A workspace whose domain objects to uninitialize.
unassign-templates	True to unassign generic scheme templates, false otherwise.
client	The client, typically a G2 window.

Description

Uninitializes all domain objects on a workspace by unrelating every datapoint that is an attribute of the domain object from its source datapoints, if specified, and optionally unassigning generic scheme templates from the domain object.

grtl-assign-template-schemes

Synopsis

grtl-assign-template-schemes
(*domain-object*: class grtl-domain-object, *client*: class ui-client-item)

Argument	Description
domain-object	The domain object to configure.
client	The client, typically a G2 window.

Description

Configures a domain object by compiling every derived datapoint that is an attribute of the domain object, relating every datapoint to source items, if applicable, that is an attribute of the domain object, and assigning generic scheme templates to the domain object.

grtl-get-scheme-by-id-for-source-object

Synopsis

grtl-get-scheme-by-id-for-source-object
    (*item*: class item, *key*: text)
    -> *scheme*: item-or-value

Argument	Description
item	The source object.
key	The scheme key of the scheme.

Return Value	Description
scheme	The scheme.

Description

Returns a scheme associated with a domain object by the scheme-key of the scheme. If no scheme is found, this API returns false.

grtl-get-scheme-by-template-id-for-source-object

Synopsis

grtl-get-scheme-by-template-id-for-source-object
(*item*: class item, *key*: text)
-> *scheme*: item-or-value

Argument	Description
item	The source object.
key	The scheme key of the scheme template.

Return Value	Description
scheme	The scheme.

Description

Returns a scheme associated with a domain object by the scheme-key of the scheme template. If no scheme is found, this API returns false.

grtl-get-schemes-for-source-object

Synopsis

grtl-get-schemes-for-source-object
    (*item*: class item)
    -> *schemes*: sequence

Argument	Description
item	The source object.

Return Value	Description
schemes	The schemes.

Description

Returns a list of schemes for a source object. The sequence contains references to the schemes.

grtl-initialize-domain-item

Synopsis

grtl-initialize-domain-item
( )

Description

Initializes a domain object by compiling every derived datapoint that is an attribute of the domain object, relating every datapoint that is an attribute of the domain object to its source datapoints, if specified, and assigning generic scheme templates to the domain object.

grtl-initialize-domain-items

Synopsis

grtl-initialize-domain-items
( )

Description

Initializes all domain objects by compiling every derived datapoint that is an attribute of a domain object, relating every datapoint that is an attribute of a domain object to its source datapoints, if specified, and assigning generic scheme templates to domain objects.

grtl-is-template-for-object

Synopsis

grtl-is-template-for-object
    (*scheme-template*: class grtl-scheme-template,
    domain-object: class grtl-domain-object)
    -> *result*: truth-value

Argument	Description
scheme-template	The scheme template to test.
domain-object	The domain object.

Return Value	Description
result	True if the scheme template is assigned to the specified domain object; false otherwise.

Description

Determines if a scheme template is assigned to a domain object.

grtl-unassign-template-schemes

Synopsis

grtl-assign-template-schemes
    (*domain-object*: class grtl-domain-object, *client*: class ui-client-item)

Argument	Description
domain-object	The domain object whose template schemes to unassign.
client	The client, typically a G2 window.

Description

Unassigns all scheme templates from all domain objects.

grtl-uninitialize-domain-item

Synopsis

grtl-uninitialize-domain-items
(*wksp*: class kb-workspace, *unassign-templates*: truth-value,
client: class ui-client-item)

Argument	Description
wskp	The workspace whose domain objects to uninitialize.
unassign-templates	Whether to unassign scheme templates as well as uninitializing domain objects.
client	The client, typically a G2 window.

Description

Uninitializes all domain objects on a workspace by unrelating every datapoint that is an attribute of the domain objects from their source datapoints, if specified, and unassigning generic scheme templates from the domain objects.

grtl-uninitialize-domain-items

Synopsis

grtl-uninitialize-domain-items
(*wksp*: class kb-workspace, *unassign-templates*: truth-value,
client: class ui-client-item)

Argument	Description
wskp	The workspace whose domain objects to uninitialize.
unassign-templates	Whether to unassign scheme templates as well as uninitializing domain objects.
client	The client, typically a G2 window.

Description

Uninitializes all domain objects on a workspace by unrelating every datapoint that is an attribute of the domain objects from their source datapoints, if specified, and unassigning generic scheme templates from the domain objects.

grtl-is-domain-object-available

Synopsis

grtl-is-domain-object-available
    (*target*: class grtl-domain-object)
    -> *result*: truth-value

Argument	Description
target	The target domain object.

Return Value	Description
result	True if the availability-status of the domain object is available; false otherwise.

Description

Determines whether a domain object is available, based on the availability-status of the domain object. Note that if a domain object in a superior process map is not available, the domain objects in the lower-level process maps are not available either.

Scheme Classes and Methods

Classes

grtl-scheme
grtl-scheme-template
grtl-scheme-template-class

Methods

grtl-scheme-template::grtl-create-scheme-for-domain-object
grtl-scheme-template::grtl-get-template-key
grtl-scheme-template::grtl-is-template-for-object
grtl-scheme::grtl-activate
grtl-scheme::grtl-deactivate
grtl-scheme::grtl-delete-source-object
grtl-scheme::grtl-evaluate
grtl-scheme::grtl-evaluate-and-return-result
grtl-scheme::grtl-get-source-object
grtl-scheme::grtl-get-value
grtl-scheme::grtl-initialize
grtl-scheme::grtl-is-scheme-active
grtl-scheme::grtl-set-source-object
grtl-scheme::grtl-set-value
grtl-scheme::grtl-validate

grtl-scheme

An abstract class that is the superior class of all advanced control schemes, implemented either as a G2 procedure or using a graphical language. For example, a GEDP diagram template is a subclass of GRTL scheme, implemented using a graphical language. A scheme defines the grtl-evaluate method, which any class that inherits from this class should implement to provide the evaluation logic. Schemes are event sources that can be activated, deactivated, and evaluated, based on event-driven changes or based on a schedule. Only active schemes can be evaluated.

A scheme can have various triggers that determine when to activate, deactivate, and evaluate the scheme. To specify these triggers, you use the evaluation-trigger, activation-trigger, and deactivation-trigger attributes, which contain structures with the following content. If not used, these attributes should contain the symbol unspecified. Note that if operand-key contains a key, then the value of the client is compared with the value of the server specified in operand-key instead of the fixed operand-value. These keys allow you to enable, disable, or evaluate the scheme, based on conditions.

The data value of the evaluation, activation, or deactivation trigger is a follows:

structure
(KEY: text, {the key of the datapoint that will trigger this scheme}
ACTIVE: truth-value {true if condition should be evaluated}
OPERATOR: text {the operator to apply}
OPERAND-KEY: text {the key of the datapoint to use as an operand}
OPERAND-VALUE: value {any valid value used as the operand if
the operand-key does not refer to a valid datapoint})

Operator is one of the following, where *datapoint-value* is the value retrieved from key in the structure above, and *operand-value* can be operand-value or the value retrieved from operand-key in the above structure:

true {Trigger condition is always true}

= {*datapoint-value* = *operand-value*}

> {*datapoint-value* > *operand-value*}

< {*datapoint-value* < *operand-value*}

>= {*datapoint-value* >= *operand-value*}

<= {*datapoint-value* <= *operand-value*}

/= {*datapoint-value* /= *operand-value*}

A scheme dispatches the following events:

- scheme-activated: Whenever the scheme is activated (no arguments).
- scheme-deactivated: Whenever the scheme is deactivated (no arguments).
- scheme-value-changed: Whenever scheme value is changed, where the value is the first argument of the event. Use grtl-set-value and grtl-get-value to change the scheme value.

Class Inheritance Path

grtl-scheme, grtl-event-source, grtl-item, item

Attributes

Attribute	Description
scheme-key	The key for referencing the scheme.
Allowable values:	Any text
Default value:	""
scheme-version	The revision number for the scheme, used to track changes.
Allowable values:	Any float
Default value:	0.0
scheme-category	A category for the scheme.
Allowable values:	Any text
Default value:	""
scheme-type	Use for organizing schemes, based on their event-detection logic.
Allowable values:	One of these symbols: general, detection, test, or response
Default value:	detection
scheme-active	Whether the scheme is active.
Allowable values:	Any truth-value

Default value: false

scheme-valid Whether the scheme is valid.

Allowable values: Any truth-value

Default value: false

evaluation-interval The frequency with which to evaluate the scheme when evaluating, based on a schedule.

Allowable values: Any item or value

Default value: 0

evaluation-counter The number of time the scheme has been evaluated.

Allowable values: Any value

Default value: 0

evaluation-repeat-count-enabled When true, the scheme is evaluated the number of times specified by evaluation-repeat-count.

Allowable values: Any truth-value

Default value: false

evaluation-repeat-count The number of times to repeat the evaluation when evaluation-repeat-count-enabled is true.

Allowable values: Any integer

Default value: 0

repeated-evaluation-countdown The number of evaluations that are left when evaluation-repeat-count-enabled is true.

Allowable values: Any integer

Default value: 0

evaluation-trigger A structure that specifies a datapoint as a trigger for evaluating the scheme. The evaluation trigger is invoked when the datapoint trigger receives a new value. The evaluation trigger compares the datapoint used as a trigger against another datapoint value. If the evaluation trigger evaluates to true, then the scheme evaluates.

Allowable values: Any value

Default value: UNSPECIFIED

activation-trigger A structure that specifies a datapoint as a trigger for activating the scheme. The activation trigger is invoked when the datapoint trigger receives a new value. The activation trigger compares the datapoint used as a trigger against another datapoint value. If the activation trigger evaluates to true, then the scheme becomes active. Only active schemes can be evaluated.

Allowable values: Any value

Default value: UNSPECIFIED

deactivation-trigger A structure that specifies a datapoint as a trigger for deactivating the scheme. The deactivation trigger is invoked when the datapoint trigger receives a new value. The deactivation trigger compares the datapoint used as a trigger against another datapoint value. If the deactivation trigger evaluates to true, then the scheme becomes inactive.

Allowable values: Any value

Default value: UNSPECIFIED

Methods

grtl-scheme::grtl-evaluate
grtl-scheme::grtl-activate
grtl-scheme::grtl-deactivate
grtl-scheme::grtl-validate
grtl-scheme::grtl-is-scheme-active
grtl-scheme::grtl-initialize
grtl-scheme::grtl-get-value
grtl-scheme::grtl-set-value
grtl-scheme::grtl-get-source-object
grtl-scheme::grtl-set-source-object
grtl-scheme::grtl-delete-source-object
grtl-scheme::grtl-evaluate-and-return-result

grtl-scheme-template

A generic scheme that is defined for a class.

Class Inheritance Path

grtl-scheme-template, item

Attributes

Attribute	Description
template-key	The key for referencing the scheme template.
Allowable values:	Any text
Default value:	""
template-version	The revision number for the scheme template, used to track changes.
Allowable values:	Any float
Default value:	0.0
template-category	A category for the scheme template.
Allowable values:	Any text
Default value:	""
template-target-class	The target class to which all instances of the scheme template applies.
Allowable values:	Any symbol
Default value:	NONE
scheme-active	Whether the scheme template is active.
Allowable values:	Any truth-value
Default value:	false
scheme-type	Use for organizing schemes, based on their event-detection logic.

Allowable values: One of these symbols:

general, detection, test, or response

Default value: detection

Methods

grtl-scheme-template::grtl-create-scheme-for-domain-object
grtl-scheme-template::grtl-get-template-key
grtl-scheme-template::grtl-is-template-for-object

grtl-scheme-template-class

A class definition for defining scheme templates.

Class Inheritance Path

grtl-scheme-template-class, grtl-class-definition, class-definition, definition, item

Attributes

Attribute	Description
class-description	See grtl-class-definition.
Allowable values:	inherited
Default value:	""
instance-key-attribute-name	See grtl-class-definition.
Allowable values:	inherited
Default value:	UUID
class-attribute-properties	See grtl-class-definition.
Allowable values:	inherited
Default value:	sequence ()

grtl-scheme-template::grtl-create-scheme-for-domain-object

Synopsis

grtl-scheme-template::grtl-create-scheme-for-domain-object
(*scheme-template*: grtl-scheme-template,
domain-object: grtl-domain-object, *client*: ui-client-item)
-> *scheme*: class grtl-scheme

Argument	Description
scheme-template	The scheme to use as a template when creating a scheme for the specified domain object.
domain-object	The domain object for which to create a scheme.
client	The client, typically a G2 window.

Return Value	Description
scheme	The new scheme.

Description

Creates a scheme for a domain object, based on a template.

grtl-scheme-template::grtl-get-template-key

Synopsis

grtl-scheme-template::grtl-get-template-key
    (*scheme-template*: grtl-scheme-template)
    -> *key*: text

Argument	Description
scheme-template	The scheme template whose key to get.

Return Value	Description
key	The key of the specified scheme template.

Description

Returns the key of a scheme template.

grtl-scheme-template::grtl-is-template-for-object

Synopsis

grtl-scheme-template::grtl-is-template-for-object
(*scheme-template*: grtl-scheme-template, *domain-object*: grtl-domain-object)
-> *return-value*: truth-value

Argument	Description
scheme-template	The scheme template to test.
domain-object	The domain object to test.

Return Value	Description
return-value	True if the specified domain object is associated with the specified template scheme.

Description

Determines whether a scheme template is associated with a domain object.

grtl-scheme::grtl-activate

Synopsis

grtl-scheme::grtl-activate
(*scheme*: grtl-scheme, *datapoint*: item-or-value, *client*: ui-client-item)

Argument	Description
scheme	The scheme to activate.
datapoint	The datapoint that triggers the activation, if applicable
client	The client, typically a G2 window.

Description

Sets the scheme-active attribute of a scheme to true and starts the periodic evaluation of the scheme, assuming the evaluation interval is greater than zero. Note that once the evaluation has started, changing the evaluation interval has no effect. To change the evaluation interval, you must deactivate and reactivate the scheme. This method dispatches the notification event scheme-activated to listeners.

grtl-scheme::grtl-deactivate

Synopsis

grtl-scheme::grtl-deactivate
    (*scheme*: grtl-scheme, *datapoint*: item-or-value, *client*: ui-client-item)

Argument	Description
scheme	The scheme to deactivate.
datapoint	The datapoint that triggers the deactivation, if applicable.
client	The client, typically a G2 window.

Description

Sets the scheme-active attribute to false and stops periodic evaluation for the scheme. This method dispatches the notification event scheme-deactivated to listeners.

grtl-scheme::grtl-delete-source-object

Synopsis

grtl-scheme::grtl-delete-source-object
(*scheme*: grtl-scheme)

Argument	Description
scheme	The scheme whose source object to delete.

Description

Deletes the reference to the current source object associated with a scheme.

grtl-scheme::grtl-evaluate

Synopsis

grtl-scheme::grtl-evaluate
    (*scheme*: grtl-scheme, *datapoint*: item-or-value, *client*: ui-client-item)

Argument	Description
scheme	The scheme to evaluate.
datapoint	The datapoint that triggers the evaluation, if applicable.
client	The client, typically a G2 window.

Description

Evaluates a scheme. You must implement this method for subclasses to evaluate instances. This method can either implement the logic of the scheme or can rely on a graphical language receiving input, for example, from plant data or alarm events.

grtl-scheme::grtl-evaluate-and-return-result

Synopsis

grtl-scheme::grtl-evaluate-and-return-result
    (*scheme*: class grtl-scheme)
    -> *result*: value

Argument	Description
scheme	The scheme to evaluate.

Return Value	Description
result	The result of evaluating the scheme.

Description

Evaluates a scheme and returns the result. If the scheme is not active, it is activated for the duration of the execution.

grtl-scheme::grtl-get-source-object

Synopsis

grtl-scheme::grtl-get-source-object
    (*scheme*: class grtl-scheme)
    -> *source-object*: item-or-value

Argument	Description
scheme	The scheme whose source object to get.

Return Value	Description
source-object	The source object associated with the specified scheme.

Description

Gets the current source object associated with a scheme. You associate a source object with a scheme to integrate the scheme with external data sources. For example, for event-driven evaluation, you would associate a source object with a scheme to evaluate the scheme when external datapoint values change.

grtl-scheme::grtl-get-value

Synopsis

grtl-scheme::grtl-get-value
    (*scheme*: class grtl-scheme)
    -> *current-value*: value

Argument	Description
scheme	The scheme whose value to get.

Return Value	Description
current-value	The current value of the specified scheme.

Description

Gets the current value of a scheme. This value can be used for input or as an output of the scheme. Evaluating the scheme may cause this value to update. For example, a GEDP diagram can use its _grtl-value attribute to store information about the execution of its blocks. Typically, this attribute is used in conjunction with a GEDP Return block, which copies the last propagated value into the value attribute of the diagram. This features allows you to use a diagram as a procedure for evaluating a scheme.

grtl-scheme::grtl-initialize

Synopsis

grtl-scheme::grtl-initialize
    (*scheme*: class grtl-scheme, *client*: ui-client-item)

Argument	Description
scheme	The scheme to initialize.
client	The client, typically a G2 window.

Description

Initializes a scheme, which relates datapoints that trigger evaluation, and enables or disables any activation, deactivation, or evaluation trigger. Trigger specification are specified on the scheme in the attributes grtl-evaluation-trigger, grtl-activation-trigger, or grtl-deactivation-trigger. For information about configuring the trigger specifications, see grtl-scheme.

grtl-scheme::grtl-is-scheme-active

Synopsis

grtl-scheme::grtl-is-scheme-active
    (*scheme*: class grtl-scheme)
    -> *active*: truth-value

Argument	Description
scheme	The scheme to test.

Return Value	Description
active	True if the scheme is active, false otherwise.

Description

Returns the value of the scheme-active attribute of a scheme. You must implement this method to return whether or not a scheme is active for advanced control schemes of the class you define.

grtl-scheme::grtl-set-source-object

Synopsis

grtl-scheme::grtl-set-source-object
    (*scheme*: class grtl-scheme, *source-object*: object)

Argument	Description
scheme	The scheme whose source object set.
source-object	The source object to associate with the scheme.

Description

Associates a source object, typically a domain object, with a scheme. Source objects are useful when entry points need to receive values in an event-driven way and you need to associate the execution of a diagram with a specific domain object.

grtl-scheme::grtl-set-value

Synopsis

grtl-scheme::grtl-set-value
(*scheme*: class grtl-scheme, *new-value*: value)

Argument	Description
scheme	The scheme whose value to set.
new-value	The new value.

Description

Sets the current value of the scheme and dispatches the notification event scheme-value-changed. The argument of the notification event is the value of the scheme. This value can be used for input or as an output of the scheme. Evaluating the scheme may cause this value to update. For example, a GEDP diagram can use its _grtl-value attribute to store information about the execution of its blocks. Typically, this attribute is used in conjunction with a GEDP Return block, which copies the last propagated value into the value attribute of the diagram. This features allows you to use a diagram as a procedure for evaluating a scheme.

grtl-scheme::grtl-validate

Synopsis

grtl-scheme::grtl-validate
(*scheme*: grtl-scheme, *client*: ui-client-item)
-> *validated*: truth-value

Argument	Description
scheme	The scheme to validate.
client	The client, typically a G2 window.

Return Value	Description
validated	True if the scheme is valid, false otherwise.

Description

Validates the scheme and dispatches the notification event scheme-validated. The argument of the notification is the result of the validation (true or false). The default implementation sets the scheme-valid attribute to true unconditionally. You must implement this method to validate an advanced control scheme of the class you define.

36

XML Serialization

Describes methods for XML serialization of G2 objects.

Introduction

GRTL includes APIs to convert XML text to/from an XML element value. The XML element value representation used in GRTL relies on the XML element value used by the G2 Web Services system procedures, which has the advantage of using a structured representation that supports large XML structures.

Specifically, an XML element value is a structure representing an XML element with this syntax:

structure
(tag-name: *text*,
attributes: *structure*,
children: *sequence*)

where:

- tag-name is the element tag name. This attribute is required.
- attributes is a structure containing named attribute values, which are texts. This attribute is optional.
- children is a sequence of XML elements and/or texts. This attribute is optional.

Attribute names use the same correspondence between XML names and G2 symbols used by G2GL, for example, *myAttribute* becomes my-attribute.

For example, this XML text:

```
<elt attrName="attrValue"><child>text1</child>text2</elt>
```

corresponds to this XML element value:

```
structure
    (tag-name: "elt",
    attributes: structure (attr-name: "attrValue"),
    children: sequence (structure (tag-name: "child",
                                   children: sequence ("text1"),
                               "text2")
```

Methods and Procedures

Methods

item::grtl-deserialize-from-xml
item::grtl-serialize-to-xml
item::grtl-serialize-to-xml

Procedures

grtl-serialize-sequence-to-xml
grtl-text-to-xml-values
grtl-file-to-xml-values
grtl-xml-values-to-object
grtl-xml-values-to-text

item::grtl-deserialize-from-xml

Synopsis

item::grtl-deserialize-from-xml
(*target*: class item, *xml-element*: structure, *xml-declaration-options*: structure)

Argument	Description
target	The object to update.
xml-element	XML element value.
xml-declaration-options	XML optional declaration information such as version, encoding, standalone, for example.

Description

This method deserializes the content of an XML element value into the attribute of the target object. Developers can implement methods for subclasses to return custom XML element value representations or deal with cases not supported by the basic methods.

item::grtl-serialize-to-xml

Synopsis

item::grtl-serialize-to-xml
(*target*: class item, *property-names-to-exclude*: sequence)
-> *xml-element*: structure, *xml-declaration-options*: structure

Argument	Description
target	The item to serialize.
property-names-to-exclude	Properties names to exclude from serialization.

Return Value	Description
xml-element	XML element value.
xml-declaration-options	XML optional declaration information such as version, encoding, standalone, for example.

Description

This method converts the user-defined attributes of an item to an XML element value representation. The default implementation converts the attribute value to a text representation in the XML structure based on the data type/property info for each attribute (see grtl-get-property-type-info). Developers can implement methods for subclasses to return custom XML element value representations or deal with cases not supported by the basic methods. This method uses the class name of the target item as the node tag name.

item::grtl-serialize-to-xml

Synopsis

item::grtl-serialize-to-xml
(*target*: class item, *property-names-to-exclude*: sequence,
node-tag-name: text)
-> *xml-element*: structure, *xml-declaration-options*: structure

Argument	Description
target	The item to serialize.
property-names-to-exclude	Properties names to exclude from serialization.
node-tag-name	The tag name of the node to serialize.

Return Value	Description
xml-element	XML element value.
xml-declaration-options	XML optional declaration information such as version, encoding, standalone, for example.

Description

This method converts the user-defined attributes of an item to an XML element value representation. The default implementation converts the attribute value to a text representation in the XML structure based on the data type/property info for each attribute (see grtl-get-property-type-info). Developers can implement methods for subclasses to return custom XML element value representations or deal with cases not supported by the basic methods. This method allows the caller to specify the node tag name.

grtl-serialize-sequence-to-xml

Synopsis

grtl-serialize-sequence-to-xml
(*sequence-of-items-or-values*: sequence,
property-names-to-exclude: sequence, *root-node-tag-name*: text,
item-node-tag-name: text, *value-node-tag-name*: text)
-> *xml-element*: structure, *xml-declaration-options*: structure

Argument	Description
sequence-of-items-or-values	The entries to serialize.
property-names-to-exclude	Properties names to exclude from serialization.
item-node-tag-name	The tag name of items within the sequence. If this is an empty string, the class name of the item is used as the tag.
value-node-tag-name	The XML tag name for entries in *sequence-of-items-or-values* that are values instead of items.

Return Value	Description
xml-element	XML element value.
xml-declaration-options	XML optional declaration information such as version, encoding, standalone, for example.

Description

This procedure converts a sequence of values or items into an XML element value representation.

grtl-text-to-xml-values

Synopsis

grtl-text-to-xml-values
    (*xml-text*: text)
    -> *xml-element*: structure

Argument	Description
xml-text	The XML text to convert.

Return Value	Description
xml-element	XML element value.

Description

This procedure converts an XML text representation to a XML element value representation.

grtl-file-to-xml-values

Synopsis

grtl-file-to-xml-values
    (*filename*: text)
    -> *xml-element*: structure

Argument	Description
filename	The file name that contains the XML text to convert.

Return Value	Description
xml-element	XML element value.

Description

This procedure imports an XML text representation from a file and returns it as an XML element value representation.

grtl-xml-values-to-object

Synopsis

grtl-xml-values-to-object
    (*xml-element*: structure)
    -> *item*: class item

Argument	Description
xml-element	XML element value.

Return Value	Description
item	The item created from the XML element.

Description

This procedure creates or updates an object based on the specification in the XML element value representation. It uses the name, uuid, or key attribute/key value to locate an existing object, and, if none can be found, creates one based on the class name specified in the XML. See item::grtl-serialize-to-xml for encoding. The child nodes are used to update attributes of the target object. This procedure signals an error if no item is found and none can be created, for example, if the class name is not specified.

grtl-xml-values-to-text

Synopsis

grtl-xml-values-to-text
(*xml-element*: structure, *xml-declarations*: structure, *xml-text*: text, *indentation*: integer)
-> *xml-text*: text

Argument	Description
xml-element	XML element value.
xml-declaration-options	XML optional declaration information such as version, encoding, standalone, for example.
xml-text	The XML text.

Return Value	Description
xml-text	The XML text.

Description

This procedure converts an XML element value to a text representation.

A

GRTL-UTILS Module

Lists the classes, procedures, methods, and functions in the grtl-utils.kb module.

Introduction

This chapter lists the classes, procedures, methods, and functions in the *grtl-utils.kb* module. All icon definitions are in *grtl-icons.kb*. All other definitions are in *grtl.kb*.

Classes in Grtl-Utils

grtl-attribute-property-details
grtl-attribute-property-info
grtl-class-definition
grtl-distribution-mode
grtl-event-source
grtl-event-source-class-definition
grtl-item
grtl-module-settings
grtl-object
grtl-permanent-item-array

grtl-permanent-item-list
grtl-vector

Procedures and Methods in Grtl-Utils

class-definition::grtl-add-event-listener
class-definition::grtl-get-item-by-key
class-definition::grtl-get-item-by-key-with-existence-check
class-definition::grtl-get-key
class-definition::grtl-get-key-attribute-name
class-definition::grtl-item-with-key-exists
class-definition::grtl-remove-all-event-listeners
class-definition::grtl-remove-event-listener
g2-list::grtl-empty-list
grtl-add-module-to-hierarchy-top
grtl-add-remote-event-listener-to-class
grtl-add-remote-event-listener-to-item
grtl-add-to-localization-resources
grtl-are-sequences-identical
grtl-assign-stray-workspaces-to-top
grtl-check-sequence-inclusion
grtl-clone
grtl-command-line-option-specified
grtl-convert-list-to-sequence
grtl-convert-string-to-valid-filename
grtl-copy-list
grtl-copy-user-defined-attributes
grtl-create
grtl-create-directory
grtl-create-module
grtl-day-to-integer
grtl-delete-contents
grtl-delete-directory
grtl-delete-list-and-contents
grtl-delete-module
grtl-distribution-mode::grtl-get-float-random-value
grtl-distribution-mode::grtl-get-integer-random-value
grtl-duration-to-g2-interval
grtl-english-ordination
grtl-evaluate-query-for-object
grtl-event-source::grtl-add-attribute-change-event-listener
grtl-event-source::grtl-add-event-listener
grtl-event-source::grtl-disable-event-notification
grtl-event-source::grtl-dispatch-event
grtl-event-source::grtl-enable-event-notification
grtl-event-source::grtl-is-event-notification-enabled

grtl-event-source::grtl-remove-all-event-listeners
grtl-event-source::grtl-remove-attribute-event-listener
grtl-event-source::grtl-remove-event-listener
grtl-extract-directory
grtl-extract-filename
grtl-extract-filename-with-extension
grtl-file-to-xml-values
grtl-filename-to-pathname
grtl-format-filename
grtl-format-symbol-to-text
grtl-g2-day-integer-to-iso-day
grtl-g2-interval-to-duration
grtl-g2-interval-to-xml-days
grtl-g2-interval-to-xml-weeks
grtl-g2-month-integer-to-iso-month
grtl-g2-time-from-text-mm-dd-yy-hh-mm-ss
grtl-g2-time-from-text-yyyy-dd-mm-hh-mm-ss
grtl-g2-time-to-iso-day
grtl-g2-time-to-iso-month
grtl-g2-time-to-iso-year
grtl-g2-time-to-iso-year-month
grtl-g2-time-to-isox-date
grtl-g2-time-to-isox-date-time
grtl-g2-time-to-isox-month-day
grtl-g2-time-to-isox-time
grtl-g2-time-to-text
grtl-g2-year-integer-to-iso-year
grtl-generate-list-of-subclasses
grtl-get-application-directory
grtl-get-command-line-option
grtl-get-default-directory
grtl-get-file-system-delimiter
grtl-get-installation-directory
grtl-get-list-of-attributes-for-class-and-subclasses
grtl-get-localization-file
grtl-get-module-by-name
grtl-get-name-of-item
grtl-get-name-or-key-of-item
grtl-get-name-or-key-of-top-level-item
grtl-get-properties-for-class
grtl-get-random-beta-value
grtl-get-random-erlang-value
grtl-get-random-exponential-value
grtl-get-random-gamma-value
grtl-get-random-lognormal-value
grtl-get-random-normal-value
grtl-get-random-triangular-value

grtl-get-random-uniform-value
grtl-get-random-value-from-text-specification
grtl-get-random-weibel-value
grtl-get-top-level-item
grtl-interval-in-hours-from-text-hh-mm-ss
grtl-interval-in-seconds-from-text-hh-mm
grtl-interval-in-seconds-from-text-hh-mm-ss
grtl-invert-sequence
grtl-is-superclass
grtl-is-valid-relation
grtl-isox-date-time-to-g2-time
grtl-isox-date-to-g2-time
grtl-load-localization-resources
grtl-localize-message
grtl-localize-message-for-client
grtl-localize-message-using-group
grtl-localize-message-without-substitutions
grtl-localize-message-without-substitutions-using-resource-group
grtl-localize-pathname
grtl-localize-pathname-and-create-directory
grtl-merge-lists
grtl-merge-localization-resources-from-file
grtl-minutes-to-hh-mm
grtl-minutes-to-hhmm
grtl-module-exists-for-file
grtl-one-key-ascending-sort
grtl-one-key-descending-sort
grtl-pad-or-truncate-string
grtl-purge-duplicates-from-list
grtl-quicksort-list
grtl-remove-cr-from-text
grtl-remove-duplicate-values-from-sequence
grtl-remove-enclosing-quotes-from-text
grtl-remove-module-references
grtl-remove-object-from-sequence
grtl-remove-remote-event-listener-to-class
grtl-remove-remote-event-listener-to-item
grtl-remove-subset-list
grtl-remove-value-from-sequence
grtl-rename-module
grtl-replace-module-references
grtl-replace-text
grtl-replace-value-in-sequence
grtl-save-localization-resources
grtl-search-and-replace-character
grtl-serialize-sequence-to-xml
grtl-set-module-file-name

grtl-sort-sequence
grtl-strip-leading-and-trailing-spaces-from-text
grtl-strip-leading-spaces-from-text
grtl-strip-trailing-spaces-from-text
grtl-text-date-and-time-from-g2-time
grtl-text-date-and-time-to-g2-time
grtl-text-date-from-g2-time
grtl-text-time-from-g2-time
grtl-text-to-g2-duration
grtl-text-to-xml-values
grtl-time-in-hours-from-text-mm-dd-yyyy-hh-mm-ss
grtl-tokenize-csv-record
grtl-unpad-string
grtl-uppercase-symbol
grtl-vector::grtl-add-element
grtl-vector::grtl-delete-all-elements
grtl-vector::grtl-delete-first-element
grtl-vector::grtl-delete-last-element
grtl-vector::grtl-get-element-at
grtl-vector::grtl-get-head-element
grtl-vector::grtl-get-size
grtl-vector::grtl-number-of-elements
grtl-vector::grtl-remove-all-elements
grtl-vector::grtl-remove-element
grtl-vector::grtl-remove-first-element
grtl-vector::grtl-remove-last-element
grtl-vector::grtl-set-size
grtl-xml-days-to-g2-interval
grtl-xml-values-to-object
grtl-xml-values-to-text
grtl-xml-weeks-to-g2-interval
item::grtl-can-delete
item::grtl-can-rename
item::grtl-confirmed-set-value
item::grtl-constructor
item::grtl-copy-attributes
item::grtl-delete
item::grtl-deserialize-from-xml
item::grtl-destructor
item::grtl-disable-event-notification
item::grtl-dispatch-event
item::grtl-enable-event-notification
item::grtl-event-notification
item::grtl-get-category
item::grtl-get-hostname-or-ip-address
item::grtl-get-item-and-property-for-pathname
item::grtl-get-item-by-key

item::grtl-get-item-by-key-with-existence-check
item::grtl-get-item-properties
item::grtl-get-key
item::grtl-get-key-attribute-name
item::grtl-get-properties
item::grtl-get-properties-type-info
item::grtl-get-property-type-info
item::grtl-get-property-value
item::grtl-get-system-info-or-best-practice-url
item::grtl-get-units
item::grtl-get-uuid
item::grtl-get-value
item::grtl-get-values-of-properties
item::grtl-get-values-of-properties-as-structure
item::grtl-initialize
item::grtl-is-clonable
item::grtl-is-event-notification-enabled
item::grtl-is-item-private
item::grtl-item-with-key-exists
item::grtl-serialize-to-xml
item::grtl-set-key
item::grtl-set-property-value
item::grtl-set-value
item::grtl-set-value-with-collection-time
item::grtl-set-values-of-properties
item::grtl-set-values-of-properties-as-structure
system-table::grtl-can-delete
system-table::grtl-can-rename
system-table::grtl-is-clonable
variable-or-parameter::grtl-calculate-rate-of-change
variable-or-parameter::grtl-calculate-standard-deviation

Functions in Grtl-Utils

comma-delimit
e
grtl-greater
grtl-mmddyyyyhhmmss-text-to-time
grtl-text-time-to-g2-time
grtl-timestamp-to-mmddhhm
grtl-unix-time-to-vb-date
pad-to-2-digits
pad-to-3-digits
pi
right-align

Index

H

I

K

L

M

N

O

P

R

S

T

U

V

W

X