G2 Error Handling Foundation

User's Guide

Version 2.3 Rev. 0

G2 Error Handling Foundation User's Guide Version 2.3 Rev. 0

May 2007

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

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

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

Part Number: DOC008-230

Contents

Preface

Describes this guide and the conventions that it uses.

About this Guide

This guide describes the G2 Error Handling Foundation (GERR) module. This module provides enhanced error handling as an extension to G2 error and G2 Foundation Resources (GFR) error handling.

Audience

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

Conventions

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

Typographic

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

Convention Examples	Description
`c:\Program Files\Gensym\`	Windows pathnames
`/usr/gensym/g2/kbs`	UNIX pathnames
`spreadsh.kb`	File names
`g2 -kb top.kb`	Operating system commands

`public void main()`
`gsi_start`
	Java, C and all other external code

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

Procedure Signatures

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

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

Related Documentation

G2 Core Technology

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

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

G2 Utilities

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

G2 Developers' Utilities

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

Bridges and External Systems

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

G2 JavaLink

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

G2 Diagnostic Assistant

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

Customer Support Services

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

To obtain customer support online:

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

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

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

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

➔ Use the following numbers and addresses:

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

Introduction to the G2 Error Handling Foundation

Describes the G2 Error Handling Foundation (GERR) module, which provides extensions to G2 and GFR error handling.

Introduction

G2 Error Handling Foundation (GERR) provides enhanced error handling as an extension to G2 error and G2 Foundation Resources (GFR) error handling.

This module assumes you are familiar with G2 and GFR error handling. For details, see the *G2 Reference Manual* and *G2 Foundation Resources User's Guide*.

When an error is signalled in G2, it causes the thread to jump up the call stack until an error handler catches it or an on error statement catches it. You can use the GERR class gerr-error, or a subclass, to define your own errors. You signal an error by creating an error object by calling the gerr-get-formatted-error method and signalling this error object.

When you catch an error, you might want to signal it again, causing the thread to jump up to the next catch, or you might want to report the error. You can call the gerr-dispatch method to convert the error object to a gerr-internal-error-alert and dispatch the alert. The dispatched alert is then handled by a GFR communication handler, which reports the alert in some way.

In some cases, your code might implement recovery actions when catching errors that should not typically be displayed. It this case, you might still want to include a way to report errors for developers to better analyze and develop their code while using predefined libraries. In such cases, you should call the gerr-dispatch-for-developers-and-delete method to log the error without aborting your code. Setting the dispatch-developer-errors attribute of the active gerr-module-settings object to true displays and logs those errors as well.

The default communication handler of GERR is to collect errors and log them to a text file so that they are available, even after deleted them in G2. These log files include files that exceed the maximum logbook pages. Depending on the system configuration, the default error handler also posts the error on the G2 Message Board. If GEVM is merged into your application, you can configure it to create an operator messages for errors that are signaled.

If you set up your application properly, you rarely need to catch errors within your procedure code. Typically, you do this only in areas of your code that must complete without aborting. For example, in GFR startup procedures, you probably want to catch all errors, dispatch them so they are not ignored, and continue processing.

Best Practices

The following sections described best practices for error handling, using GERR.

Using an Object-Oriented Approach for Error Handling

To use an object-oriented approach for error handling:

1 At development time, create permanent instances of gerr-error-class by creating class-definition subclasses for specific error types and for every potential type of error.

2 Assign a class name to each of these permanent error classes and configure the gfr-text-resource attribute.

Using a resource file enables you to separate the text of the error from the code, making it easier to find and update the text in a consistent manner and optionally to localize the text.

3 Add new entries to the GFR text resource object and file, using the following pattern, replacing *my-error-class* with the name of your error class and *my-error-class-description* with the description of your error:

ERROR.*my-error-class*, *my-error-class-description*

The text description supports up to nine substitutions marked [1] to [9] in the description.

4 To signal an error, use the following code:

```
{Define the error variable}
gerr-error my-error-to-signal;

{Get the error object to signal}
my-error-to-signal =
    call gerr-get-formatted-error(symbolic-name-of-error-class,
    this procedure, notes, sequence( {up to 9 substitution arguments}),
    gfr-default-window {or any g2-window});

{Signal the error}
signal my-error-to-signal;
```

Using Consistent Error Handling

The following represent best practices for different types of error handling:

- You do not catch errors, but rather, leave it to calling procedures to process errors.

  ➔ Do not use any on error statements in your code.

- You need to catch an error and do some clean up, and you want to resignal the error.

  a Use an on error statement in you code to catch the error.

  b Do the required cleanup within the on error block of code.

  c Resignal the error by using the signal statement within the on error block of code.

  Note that G2 updates the error-source-item, the error-source-line, and error-source-column attributes of the error as if the error occurred in your on error code, rather than pointing to the original procedure, line, and column where the error occurred.

- You need to catch an error, do some clean up, and provide notification about the error and/or log the error, but you do not want to interrupt the flow of your procedure.

  a Use an on error statement in you code to catch the error.

  b Do the required cleanup within the on error block of code.

  c Add a call to gerr-dispatch within the on error block of code. This call processes the error by logging it to a file, adding it to the Message Browser, and/or displaying it on the G2 Message Board, depending on GERR and GEVM configurations. By default, the last 100 errors are kept in a buffer, and the oldest errors are automatically deleted when the buffer is full.

d. Depending on the logic, after the call to gerr-dispatch or after the end of the on error statement, continue with the logic of your procedure.

- You need to catch the error, would like to hide the error, and do not want to interrupt the flow of your procedure, but you still want to notify developers about the error so they can understand and diagnose specific problems. This occurs, for example, when you expect potential errors to occur but your code includes logic for automatically recovering from the error.

  a. Use an on error statement in you code to catch the error.

  b. Do the required cleanup within the on error block of code.

  c. Add a call to gerr-dispatch-for-developers-and-delete within the on error block of code. This call processes the error if dispatch-developer-errors in the active gerr-module-settings object is true, logging it to a file, adding it to the Message Browser, and/or displaying it on the G2 Message Board, based on GERR and GEVM configurations. By default, the last 100 errors are kept in a buffer, and the oldest errors are automatically deleted when the buffer is full. See gerr-module-settings on page 8.

  d. Depending on the logic, after the call to gerr-dispatch or after the end of the on error statement, continue with the logic of your procedure.

Disabling GERR Logging

You disable GERR logging and just display errors in the G2 log book. Alternatively, you might want to create your own error handler, which allows you to perform custom error handling, such as posting the error and propagating it to the basic GERR error handler.

To disable GERR logging:

1. Create a gfr-startup-settings object in your top-level module or in module that is at a higher level than GERR in the module hierarchy.

2. Change the gfr-error-handling-enabled attribute to false.

You can also configure this attribute in the configuration file. See “Configuration File” on page 10.

Configuring GERR Logging

You can configure various aspects of GERR logging, or you can create a custom error handler.

To enable and configure GERR logging:

1 Optionally, create a gfr-startup-settings object in your top-level module or in a module that is at a higher level than GERR in the module hierarchy.

2 Configure the gfr-error-handling-enabled attribute of the active gfr-startup-settings to be true.

3 Call gerr-configure from your gfr-startup procedure to configure whether error messages should be logged to a file and/or posted to the Message Board.

Alternatively, store an instance of a gerr-module-settings in a higher level module and configure it as desired.

4 To analyze the error log file using Excel, set the error-logging-in-csv-format attribute of the active gerr-module-settings to true.

5 To display operator messages for signalled errors in addition to handling errors by calling gerr-configure, set the enable-g2-error-handler attribute of the active gevm-module-settings object to true.

To create a custom error handler:

1 Create a gfr-communication-handler in a module that is at a higher level than GERR in the module hierarchy.

2 Specify the error handler procedure.

For example:

```
my-communication-handler (communication: class gerr-internal-error-alert,
    initiating-item: item-or-value, client: class ui-client-item) = (structure)
{--- This handler is part of a SW development environment. }
response: structure;
begin
    response = call gerr-internal-error-alert-handler (communication,
        initiating-item, client);
    post "[the error-description of initiating-item]";
return structure ();
end
```

Loading GERR

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

Module Settings

Describes the G2 Error Handling Foundation (GERR) module settings.

Introduction

The gerr-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.

gerr-module-settings

Manages system configurations for the GERR module.

Class Inheritance Path

gfr-module-settings, object, item

Attributes

Attribute	Description
error-ring-size	The size of the buffer for keeping errors to be dispatched. Old errors are deleted from the ring to make room for new.
Allowable values:	integer
Default value:	100
error-logging-enabled	Whether error logging is enabled.
Allowable values:	truth-value
Default value:	true
Notes:	See “Configuration File” on page 10.
error-logging-in-csv-format	Whether to keep errors in CSV file format.
Allowable values:	truth-value
Default value:	false
error-log-file	The name of the error log file when logging is enabled.
Allowable values:	text
Default value:	kb-errors.log
Notes:	See “Configuration File” on page 10.

Attribute	Description
show-errors-in-message-board	Whether to show errors in the G2 Message Board.
Allowable values:	truth-value
Default value:	true
Notes:	See “Configuration File” on page 10.
dispatch-developer-errors	Whether error messages that should typically be discarded are logged and displayed. Set to true when developing applications to see all errors, including errors that are typically deleted by the code.
Allowable values:	truth-value
Default value:	false

Configuration File

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

Group	Configuration File Settings	GERR Module Settings Attributes/ Description
GRTL	`APPLICATION-ERROR-LOG-ENABLED=true`	error-logging-enabled
GRTL	`APPLICATION-ERROR-LOG-FILE=kb-errors.log`	error-log-file
GRTL	`APPLICATION-ERROR-INFORM-ENABLED=true`	show-errors-in-message-board
GRTL	`APPLICATION-ERROR-ENABLED=true`	Configures the gfr-error-handling-enabled attribute in the gfr-startup-settings. For more information, see “Disabling GERR Logging” on page 4.

Note These configuration file settings are imported by GRTL, not by GERR.

Classes

Describes the GERR module classes.

Introduction

G2 Error Handling Foundation (GERR) provides the following classes:

- gerr-error-class on page 12
- gerr-error on page 13
- gerr-alert on page 15
- gerr-internal-error-alert on page 16
- gerr-operator-alert on page 18

gerr-error-class

A class definition to use for specifying your own error classes. Instances of gerr-error-class define the error classes, which you use to define class hierarchies of errors for use within on error statements.

Class Inheritance Path

gerr-error-class, class-definition

Attributes

Attribute	Description
gfr-text-resource	See gfr-error in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	unspecified
gfr-message-name	See gfr-error in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	unspecified
gfr-localized-text	See gfr-error in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	""

gerr-error

The gerr-error class enhances the gfr-error class by adding support for text substitutions, error origin, and notes. Further, you can either signal GERR errors by calling the gerr-signal-error method or dispatch the error directly without aborting the current thread by calling gerr-dispatch. Note that GERR eventually dispatches GERR errors if your code or other G2 code does not catch them first.

At a minimum, you must configure the gfr-text-resource and gfr-message-name attributes. Additionally, we recommend that you name the object by using the same symbol that you provide for the gfr-message-name attribute. In this way, you can easily reference the error by name in your calls to gerr-signal-error. Also, the set of gerr-error named instances for your modules provide documentation of the different errors that your modules can signal.

If you need to resignal a gerr-error after catching it with an on error action or gfr-error-handler, you must use the G2 signal action.

Class Inheritance Path

gerr-error, gfr-error, error, gerr-object, object, gerr-item, item

Attributes

Attribute	Description
gfr-text-resource	See gfr-error in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	unspecified
gfr-message-name	See gfr-error in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	unspecified
gfr-localized-text	See gfr-error in the *G2 Foundation Resources User's Guide*.

Allowable values:	inherited
Default value:	""

Methods

gerr-error::gerr-dispatch on page 28

gerr-error::gerr-dispatch on page 29

gerr-error::gerr-get-formatted-transient-error on page 30

gerr-get-formatted-error on page 43

gerr-error::gerr-get-contextual-data on page 32

gerr-error::gerr-get-unprocessed-contextual-data on page 33

gerr-error::gerr-signal-error on page 34

gerr-error::gerr-signal-error on page 35

gerr-alert

The gerr-alert class is an abstract class superior to gerr-internal-error-alert and gerr-operator-alert.

Class Inheritance Path

gerr-alert, gfr-alert, gfr-communication, gerr-object, object, gerr-item, item

Attributes

Attribute	Description
gfr-prompt-text	See gfr-alert in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	gfr-text-proxy
gfr-button-label	See gfr-alert in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	gfr-ok-text-proxy
gfr-handler-class	See gfr-communication in the *G2 Foundation Resources User's Guide.*
Allowable values:	inherited
Default value:	gfr-alert-handler

gerr-internal-error-alert

The gerr-internal-error-alert class is a gfr-alert subclass for use in reporting internal program errors. The gerr-dispatch method creates a gerr-internal-error-alert from an error object. GFR communication handlers are passed the alert object and the initiating item, which is the error object in the case of internal error alerts. From the initiating error object, you can access programmatically the items and values specified when the error was signalled or dispatched. For more information, see gerr-error::gerr-get-contextual-data on page 32.

Class Inheritance Path

gerr-internal-error-alert, gerr-alert, gfr-alert, gfr-communication, gerr-object, object, gerr-item, item

Attributes

Attribute	Description
gfr-prompt-text	See gfr-alert in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	gfr-text-proxy
gfr-button-label	See gfr-alert in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	gfr-ok-text-proxy
gfr-handler-class	See gfr-communication in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	gfr-alert-handler
_gerr-error-source-trace	The value of the error-source-trace attribute from error instances.

Allowable values:	sequence
Default value:	sequence()

gerr-operator-alert

A gerr-operator-alert is created when you call the procedure gerr-dispatch-operator-alert. The class exists so that you can define your own GFR communication handler for operator alerts.

Class Inheritance Path

gerr-operator-alert, gerr-alert, gfr-alert, gfr-communication, gerr-object, object, gerr-item, item

Attributes

Attribute	Description
gfr-prompt-text	See gfr-alert in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	gfr-text-proxy
gfr-button-label	See gfr-alert in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	gfr-ok-text-proxy
gfr-handler-class	See gfr-communication in the *G2 Foundation Resources User's Guide*.
Allowable values:	inherited
Default value:	gfr-alert-handler

4

Methods and Procedures

Describes the GERR module methods and procedures.

Introduction

G2 Error Handling Foundation (GERR) provides a number of methods and procedures used for error handling, including signalling, dispatching, and formatting errors.

Methods

error::gerr-add-to-recent-errors

Synopsis

error::gerr-add-to-recent-errors
    (*err*: error)

Argument	Description
err	The error object to be added.

Description

This procedure is called internally whenever any error is dispatched via gerr-dispatch. The error is added to a ring buffer of the most recent errors to be dispatched. When the ring is full, the oldest error is deleted, provided it is a transient object. If your code catches a gerr-error error with the G2 on error statement, you should call this procedure to keep the error before ensuring its eventual deletion, to delete it immediately, or to pass on the responsibility by resignalling the error.

If you pass this procedure a permanent error, the error will not be deleted, which might result in a memory leak. After an error has been dispatched and before it has been deleted by this procedure, it is accessible through the GFR alert API as the initiating item of the alert.

error::gerr-dispatch

Synopsis

error::gerr-dispatch
(*err*: error)

Argument	Description
err	The error object to be dispatched.

Description

This method convert *err* into a gerr-internal-error-alert and dispatches the alert specifying the GFR default window (see note below). If *err* is a gfr-error, then the alert text is localized. If *err* is a gerr-error, then the alert text is localized with text substitutions.

If you need to specify the procedure dispatching the error, use the two-argument version of this method. If you need to specify the procedure dispatching the error and a window other than the GFR default window, use the three-argument version of this method.

Note that the default GERR communication handler does not report this alert to a window. It simply logs it to the file specified by gerr-log-file. To override or add to this behavior, you must create your own GFR communication handler.

error::gerr-dispatch

Synopsis

error::gerr-dispatch
(*err*: error, *dispatcher*: item-or-value, *win*: g2-window)

Argument	Description
err	The error object to be dispatched.
dispatcher	The procedure dispatching the error, or the symbol unspecified.
win	The G2 window to which the alert is to be dispatched.

Description

This method converts an error object into a gerr-internal-error-alert, allowing you to record the name of the procedure dispatching the error and to specify a G2 window to which the alert should be dispatched. Dispatched transient errors are eventually deleted for you. If you pass in a permanent error, a transient copy is made for you and eventually automatically deleted. For more information, see error::gerr-add-to-recent-errors on page 22.

Note that the default GERR communication handler does not report this alert to a window. It simply logs it to the file specified by gerr-log-file. To override or add to this behavior, you must create your own GFR communication handler.

error::gerr-dispatch

Synopsis

error::gerr-dispatch
(*err*: error, *dispatcher*: item-or-value)

Argument	Description
err	The error object to be dispatched.
dispatcher	The procedure dispatching the error or the symbol unspecified. For backward compatibility, specify the G2 window to which the error is to be dispatched.

Description

This method converts an error object into a gerr-internal-error-alert, allowing you to record the name of the procedure dispatching the error. Dispatched transient errors are eventually deleted for you. If you pass in a permanent error, a transient copy is made for you and eventually automatically deleted. For more information, see error::gerr-add-to-recent-errors on page 22.

For backward compatibility, the second argument can be a G2 window to which the alert should be dispatched. If you want to simply dispatch to the GFR default window without specifying a dispatching procedure, use the two-argument version of this method. If you want to specify both a dispatcher and a window, use the three-argument version of this method.

Note that the default GERR communication handler does not report this alert to a window. It simply logs it to the file specified by gerr-log-file. To override or add to this behavior, you must create your own GFR communication handler.

error::gerr-get-formatted-transient-error

Synopsis

error::gerr-get-formatted-transient-error
  (*err*: class error, *procedure*: item-or-value, *error-notes*: text,
  contextual-data: item-or-value)
  –> *error*: class error

Argument	Description
err	The error object to be formatted.
procedure	The procedure signaling the error, or, for backward compatibility, the name of the procedure.
error-notes	Any extra information relevant to this occurrence of the error.
contextual-data	An item-or-value or a sequence of item-or-values to be used as text substitutions in the GFR localization of the error. If an item is given or included in the sequence, GERR uses its name or the symbol unnamed-item if the item has no name. Structures and nested sequences cause errors.

Return Value	Description
error	The error object.

Description

This method formats an error object, localizing the error message using the gfr-text-resource of the error, and stores the localized text in the gfr-localized-text attribute of the error.

If no error-description is defined, the localized text is assigned to the G2 error message as the default text. This ensures that a text is displayed in the G2 Logbook if no GFR error handler is used. Later, a GFR communication handler might update it to include additional information, such as the source line, source column, and source item.

If the error argument to the method is permanent, the method returns a transient cloned object. Each transient error object is configured based on the specific context where the error occurs, for example, the error-description or error-source-

item. This ensures that unique error objects are signalled for specific errors if, for example, multiple errors are signaled from the same source.

gerr-error::gerr-dispatch

Synopsis

gerr-error::gerr-dispatch
(*err*: gerr-error, *proc*: item-or-value, *error-notes*: text,
contextual-data: item-or-value)

Argument	Description
err	The gerr-error to be dispatched.
proc	The procedure dispatching the error. For backward compatibility, specify the name of the procedure.
error-notes	Any extra information relevant to this occurrence of the error.
contextual-data	An item-or-value or a sequence of item-or-values to be used as text substitutions in the GFR localization of the error. If an item is given or included in the sequence, GERR uses its name or the symbol unnamed-item, if the item has no name. Providing a structure or a nested sequence causes an error.

Description

This method clones a transient error from the permanent error object; no cloning occurs if you pass a transient error. It then stores the procedure name and notes in the new error object, processes the contextual data, for example, to resolve items to their names, and reports the error. Use this method when you want to report an error with all the extra information normally provided to the gerr-signal-error method, but without actually signalling the error. The call stack will not be aborted.

To dispatch an error to a window other than the default GFR window, use the five-argument version of this method.

gerr-error::gerr-dispatch

Synopsis

gerr-error::gerr-dispatch
(*err*: gerr-error, *proc*: item-or-value, *error-notes*: text,
contextual-data: item-or-value, *win*: class g2-window)

Argument	Description
err	The gerr-error to be dispatched.
proc	The procedure dispatching the error. For backward compatibility, specify the name of the procedure.
error-notes	Any extra information relevant to this occurrence of the error.
contextual-data	An item-or-value or a sequence of item-or-values to be used as text substitutions in the GFR localization of the error. If an item is given or included in the sequence, GERR uses its name or the symbol unnamed-item, if the item has no name. Providing a structure or a nested sequence causes an error.
win	The G2 window that the GERR default error handler uses in dispatching this error.

Description

This method clones a transient error from the permanent error object; no cloning occurs if you pass a transient error. It then stores the procedure name and notes in the new error object, processes the contextual data, for example, to resolve items to their names, and reports the error. Use this method when you want to report an error with all the extra information normally provided to the gerr-signal-error method but without actually signalling the error. The call stack will not be aborted.

To dispatch an error to the default GFR window, use the four-argument version of this method.

gerr-error::gerr-get-formatted-transient-error

Synopsis

gerr-error::gerr-get-formatted-transient-error
(*err*: class gerr-error, *procedure*: item-or-value, *error-notes*: text,
contextual-data: item-or-value)
–> *error*: class gerr-error

Argument	Description
err	The error object to be added.
procedure	The procedure signaling the error, or for backward compatibility, the name of the procedure.
error-notes	Any extra information relevant to this occurrence of the error.
contextual-data	An item-or-value or a sequence of item-or-values to be used as text substitutions in the GFR localization of the error. If an item is given or included in the sequence, GERR uses its name or the symbol unnamed-item if the item has no name. Structures and nested sequences cause errors.

Return Value	Description
error	The error object.

Description

This method formats an error object, localizing the error message using a GFR text resource and storing it in the gfr-localized-text attribute of the error.

If no error-description is defined, the localized text is assigned to the G2 error message as the default text. This ensures that a text is displayed in the G2 Logbook if no GFR error handler is used. Later, a GFR communication handler might update it to include additional information, such as the source line, source column, and source item.

If the error argument to the method is permanent, the method returns a transient cloned object. Each transient error object will be configured based on the specific context where the error occurs, for example, the error description or error source

item. This ensures that unique error objects are signalled for specific errors if, for example, multiple errors are signaled from the same source.

gerr-error::gerr-get-contextual-data

Synopsis

gerr-error::gerr-get-contextual-data
(*err*: gerr-error)
–> *contextual-data*: sequence

Argument	Description
err	The gerr-error to be queried.

Return Value	Description
contextual-data	The contextual data of the error object.

Description

This method returns the contextual data of an error. This method always returns a sequence, even if a single item or value was specified for contextual data in a call to gerr-signal-error. If no contextual data exists for the error, an empty sequence is returned. The data returned is based upon the contextual data provided, if any, in the call to gerr-signal-error. GERR processes the data for use in text substitution. To return contextual data without process it, use gerr-error::gerr-get-unprocessed-contextual-data.

gerr-error::gerr-get-unprocessed-contextual-data

Synopsis

gerr-error::gerr-get-unprocessed-contextual-data
    (*err*: gerr-error)
–> *conextual-data*: sequence

Argument	Description
err	The gerr-error to be queried.

Return Value	Description
conextual-data	The contextual data of the error object.

Description

This method returns the contextual data of an error. This method always returns a sequence, even if a single item or value was specified for contextual data in a call to gerr-signal-error. If no contextual data exists for the error, an empty sequence is returned. The data returned is exactly the contextual data provided, if any, in the call to gerr-signal-error, except that single items or values are returned as a sequence of one. To return and process the contextual data for text substitution, use gerr-error::gerr-get-contextual-data.

gerr-error::gerr-signal-error

Synopsis

gerr-error::gerr-signal-error
(*err*: gerr-error, *procedure*: item-or-value, *error-notes*: text,
contextual-data: item-or-value)

Argument	Description
err	The gerr-error to be signalled.
procedure	The procedure signalling the error, or, for backward compatibility, the name of the procedure.
error-notes	Any extra information relevant to this occurrence of the error.
contextual-data	An item-or-value or a sequence of item-or-values to be used as text substitutions in the GFR localization of the error. If an item is given or included in the sequence, GERR uses its name, or the symbol unnamed-item if the item has no name. Providing a structure or nested sequence causes an error.

Description

This method clones a transient error from the permanent error object; no cloning occurs if you pass a transient error. It then stores the procedure name and notes in the new error object, processes the contextual data, for example, to resolve items to their names, and signals the error. If the default GERR error handler handles the error, the error is dispatched to the GFR default window.

If you have no need for text substitutions, use the three-argument version of this method. If you want to specify a particular G2 window for the GERR error handler to use in dispatching the error, use the five-argument version of this method.

This method calls gerr-get-formatted-transient-error to format the message content.

gerr-error::gerr-signal-error

Synopsis

gerr-error::gerr-signal-error
(*err*: gerr-error, *procedure*: item-or-value, *error-notes*: text,
contextual-data: item-or-value, *win*: class g2-window)

Argument	Description
err	The gerr-error to be signalled.
procedure	The procedure signalling the error, or, for backward compatibility, the name of the procedure.
error-notes	Any extra information relevant to this occurrence of the error.
contextual-data	An item-or-value or a sequence of item-or-values to be used as text substitutions in the GFR localization of the error. If an item is given or included in the sequence GERR will use its name or the symbol unnamed-item if the item has no name. Structures and nested sequences will cause errors.
win	The G2 window to be used by the GERR default error handler in dispatching this error.

Description

This method clones a transient error from the permanent error object; no cloning occurs if you pass a transient error. It then stores the procedure name and notes in the new error object, processes the contextual data, for example, to resolve items to their names, and signals the error. If the default GERR error handler handles the error, the error is dispatched to *win*.

To signal the error to the GFR default window, use the four-argument version of this method. If you have no need for text substitutions, use the three-argument version of this method.

This method calls gerr-get-formatted-transient-error to format the message content.

gerr-error::gerr-signal-error

Synopsis

gerr-error::gerr-signal-error
(err: *gerr-error*, *procedure*: item-or-value, *error-notes*: text)

Argument	Description
err	The gerr-error to be signalled.
procedure	The procedure signalling the error, or, for backward compatibility, the name of the procedure.
error-notes	Any extra information relevant to this occurrence of the error.

Description

This method clones a transient error from the permanent error object; no cloning occurs if you pass a transient error. It then stores the procedure name and notes in the new error object and signals the error. If the default GERR error handler handles the error, the error is dispatched to the GFR default window.

If you want to specify text substitutions to include in the GFR localization of the error, use the four-argument version of the method. If you also want to specify a particular G2 window for the GERR error handler to use in dispatching the error, use the five-argument version of this method.

This method calls gerr-get-formatted-transient-error to format the message content.

gfr-error::gerr-get-formatted-transient-error

Synopsis

gfr-error::gerr-get-formatted-transient-error
(*err*: class gfr-error, *procedure*: item-or-value, *error-notes*: text,
contextual-data: item-or-value)
–> *error*: class gfr-error

Argument	Description
err	The error object to be added.
procedure	The procedure signaling the error, or for backward compatibility, the name of the procedure.
error-notes	Any extra information relevant to this occurrence of the error.
contextual-data	An item-or-value or a sequence of item-or-values to be used as text substitutions in the GFR localization of the error. If an item is given or included in the sequence, GERR uses its name or the symbol unnamed-item if the item has no name. Structures and nested sequences cause errors.

Return Value	Description
error	The error object.

Description

This method formats an error object, localizing the error message using a GFR text resource and storing it in the gfr-localized-text attribute of the error.

If no error-description is defined, the localized text is assigned to the G2 error message as the default text. This ensures that a text is displayed in the G2 Logbook if no GFR error handler is used. Later, a GFR communication handler might update it to include additional information, such as the source line, source column, and source item.

If the error argument to the method is permanent, the method returns a transient cloned object. Each transient error object will be configured based on the specific context where the error occurs, for example, the error description or error source

item. This ensures that unique error objects are signalled for specific errors if, for example, multiple errors are signaled from the same source.

Procedures

gerr-configure

Synopsis

gerr-configure
(*post-errors*: truth-value, *enable-error-logging*: truth-value, *log-file*: text)

Argument	Description
post-errors	If true, posts errors to the Message Board.
enable-error-logging	If true, logs errors to the specified log file.
log-file	The log file in which to log errors.

Description

Configures whether to log errors to the Message Board and/or log file. By default, errors are logged to both the Message Board or to a log file. These configuration settings are used by the default error handler gerr-internal-error-alert-handler.

gerr-dispatch-for-developers-and-delete

Synopsis

gerr-dispatch-for-developers-and-delete
(*err*: class error, *dispatcher*: item-or-value)

Argument	Description
err	The error object to be dispatched.
dispatcher	The procedure dispatching the error, the symbol unspecified, or, for backward compatibility, the G2 window to which the error should be dispatched.

Description

Dispatches and deletes an error when dispatch-developer-errors is true in the gerr-module-settings object. This procedure allows you to delete errors that should be hidden in a deployed application, but allows developers to view internal errors to assist in debugging. The procedure also ensures that the error is deleted appropriately to avoid memory leaks. The *dispatcher* procedure might include recovery conditions for the error.

gerr-dispatch-operator-alert

Synopsis

gerr-dispatch-operator-alert
 (*alert-name*: symbol, *text-resource-name*: symbol,
 contextual-data: item-or-value, *win*: g2-window)

Argument	Description
alert-name	A GFR text resource key.
text-resource-name	The name of a GFR text resource.
contextual-data	A named item, simple value, or sequence of named items and simple values to be used as text substitutions.
win	The window where the GFR alert is to be sent.

Description

This procedure creates, localizes, dispatches, and deletes a GFR alert of class gerr-operator-alert. Calling this procedure replaces four separate actions required when using GFR's built-in facilities. Note that calling this procedure with gfr-default-window causes the alert to be sent to all connected windows.

gerr-get-formatted-error

gerr-get-formatted-error
(*error-class-name*: symbol, *procedure*: item-or-value, *error-notes*: text, *contextual-data*: item-or-value, *win*: item-or-value)
–> *error*: class gerr-error

Synopsis

Argument	Description
error-class-name	The class name of the error class.
procedure	The procedure signaling the error, or for backward compatibility, the name of the procedure.
error-notes	Any extra information relevant to this occurrence of the error.
contextual-data	An item-or-value or a sequence of item-or-values to be used as text substitutions in the GFR localization of the error. If an item is given or included in the sequence, GERR uses its name or the symbol unnamed-item if the item has no name. Structures and nested sequences cause errors.
win	The client.

Return Value	Description
error	The error object.

Description

Creates a gerr-error object and returns it. The error-description is formatted based on the gfr-text-resource and gfr-message-name attributes of the error. This procedure allows the definition of the message text and the code to be kept separate by loading error messages from a text file, which can be spell checked, for example, and can provide localized text.

gerr-internal-error-alert-handler

Synopsis

gerr-internal-error-alert-handler
(*communication*: gerr-internal-error-alert, *initiating-item*: item-or-value,
client: ui-client-item)
–> *response*: structure

Argument	Description
communication	The gerr-internal-error-alert object to be handled.
initiating-item	The item that caused the error.
client	The G2 client to be used by the handler in dispatching errors.

Return Value	Description
response	See gfr-communications-handler in the *G2 Foundation Resources User's Guide*.

Description

This procedure provides a built-in communications handler that handles all GERR internal error alerts generated by the built-in GERR error handler, which catches all uncaught errors. This handler writes the contents of the error communication to the error file. Other communications handlers can call this handler to combine other behavior with the default behavior of this handler.

The description in the error message includes the source line, source column, and source item. The error display and log file include the error line, error column, and error source item.

If the error-logging-in-csv-format attribute in the gerr-module-settings object is true, the error handler log file is written as a CSV file.

The handler uses us-ascii text conversion when writing error rows to the log file.

gerr-set-log-file

Synopsis

gerr-set-log-file
(*log-file*: text)

Argument	Description
log-file	A pathname to the log file to set.

Description

Sets the current log file. The default value is `logs\kb-errors.log` in the default installation directory.

Index