G2 Reporting Engine

User’s Guide

Version 2.3 Rev. 0

G2 Reporting Engine User’s Guide, Version 2.3 Rev. 0
January 2008

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: DOC012-230

Contents

Chapter 4

Charting

Chapter 5

Excel Integration

Preface

Describes this document and the conventions that it uses.

About this Guide

This guide describes the G2 Reporting Engine (GRPE) module. This module provides a consistent approach for creating reports and charting report data, and provides an Excel interface.

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 <u>*underlined*</u>. Each value is followed by its type:

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

Related Documentation

G2 Core Technology

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

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

G2 Utilities

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

G2 Developers' Utilities

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

Bridges and External Systems

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

G2 JavaLink

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

G2 Diagnostic Assistant

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

Customer Support Services

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

To obtain customer support online:

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

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

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

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

➔ Use the following numbers and addresses:

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

Introduction to the G2 Reporting Engine

Describes the G2 Reporting Engine (GRPE) module.

Introduction

G2 Reporting Engine (GRPE) defines out-of-the-box reports, classes, and APIs to define reports and charts, and dialogs to configure and visualize them. GRPE provides a consistent approach for defining reports and charts, collecting values, displaying tabular values in reports, and charting those values.

GRPE supports collecting values from CSV files, databases, or G2 items, displaying values in Telewindows Next Generation, and exporting values to CSV files, Excel, databases, and G2 items. The contents of a report can be updated on demand, using menu choices or APIs, or on a regular interval.

Chart enabled reports support defining multiple chart views.

GRPE also defines APIs for easy integration with Excel. These APIs enable you to easily configure Excel workbooks dynamically from G2. To enable this functionality, GRPE provides an Excel add-in. Once installed and included in Excel, you can connect Excel to a G2 server and receive dynamic updates from G2.

Loading GRPE

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

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

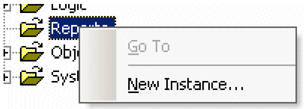
2

Built-in Reports

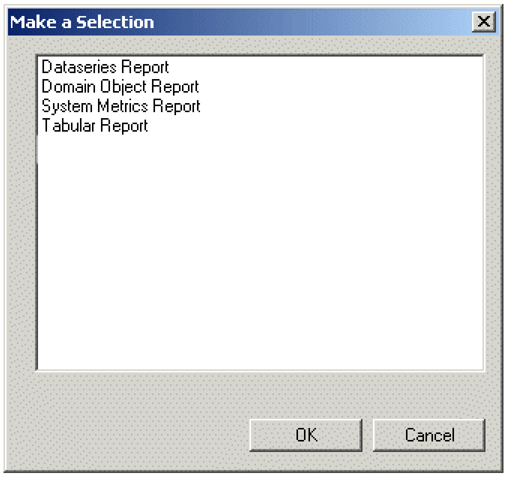
Describes the built-in reports in the G2 Reporting Engine (GRPE) module.

Introduction

GRPE provides several built-in reports. You create these reports by right clicking the Report node in the navigator and selecting the New Instance menu choice:



A dialog is displayed asking the type of report you wish to create. The list of available reports depends on the modules that are loaded. Here is the default set of reports available when GRPE is loaded:



Make a Selection

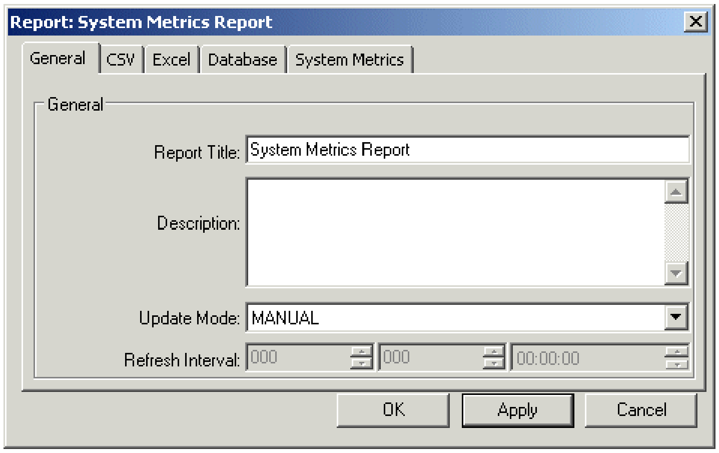
Dataseries Report
Domain Object Report
System Metrics Report
Tabular Report

In addition to these built-in reports, GEVM provides an Alarm report and an Alarm Statistics report, and BRMS provides the Rule Usage report. For more information, see the *G2 Event Manager User's Guide* and *Business Rules Management System User's Guide*.

Configuring Reports

To configure a report, you configure the update mode to be manual or based on a refresh interval. You can configure reports to export data to CSV, a database, or Excel. The Dataseries and Tabular reports also allow you to import data from CSV or a database.

Here is the General tab for the System Metrics report:



Report: System Metrics Report

General

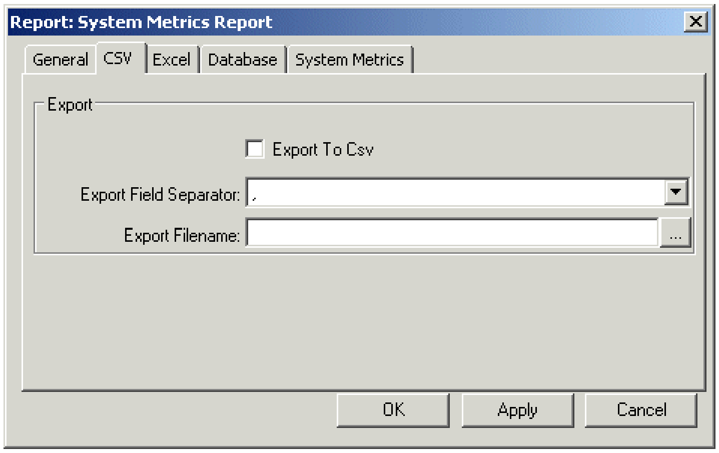
Report Title: System Metrics Report

Description:

Update Mode: MANUAL

Refresh Interval: 000 000 00:00:00

To export report data to CSV, enable Export to CSV and configure filename and separator. Here is the CSV tab for the System Metrics report:



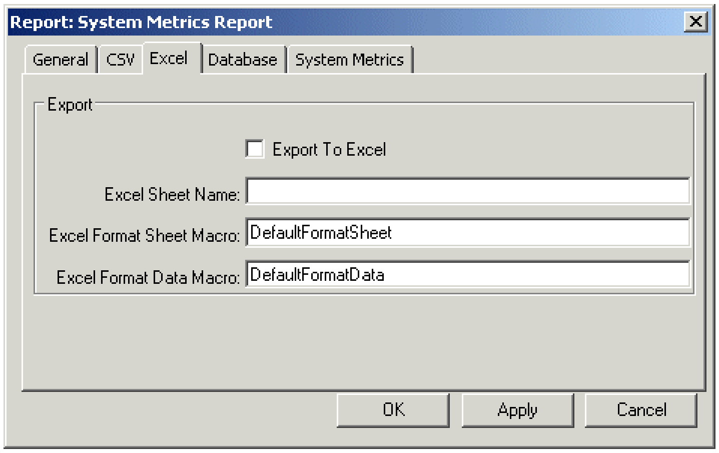
Export

☐ Export To Csv

Export Field Separator: ,

Export Filename: ...

To export data to Excel, enable the Export to Excel option and configure the sheet name. You can also configure custom macros for formatting the sheet and data. Here is the Excel tab for the System Metrics report:



Export

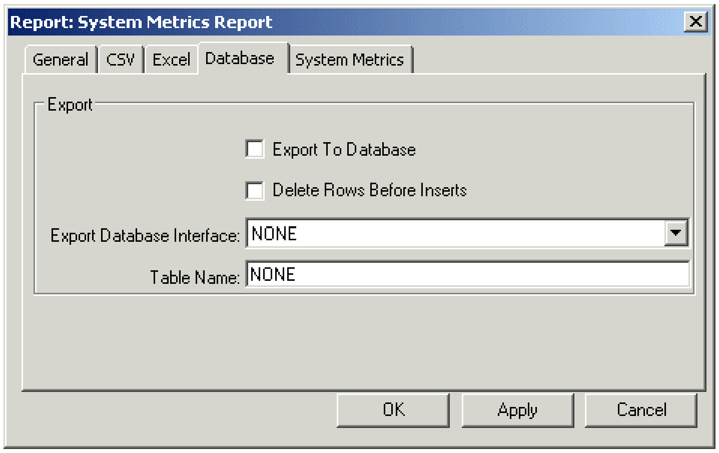
☐ Export To Excel

Excel Sheet Name:

Excel Format Sheet Macro: DefaultFormatSheet

Excel Format Data Macro: DefaultFormatData

To export data to a database, enable the Export to Database option, and configure the database interface, table name, and whether to delete rows before inserts. Here is the Database tab for the System Metrics report:



Report: System Metrics Report

☐ Export To Database

☐ Delete Rows Before Inserts

Export Database Interface: NONE

Table Name: NONE

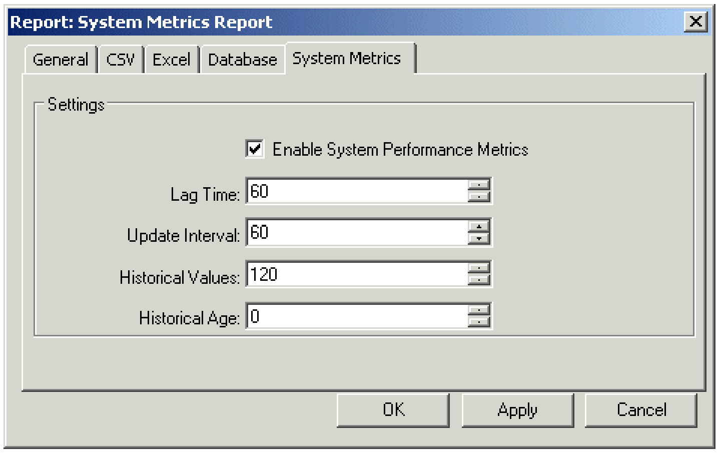
To create a database interface object, in the Navigator, expand the System Settings > Adapters node and choose New Instance on SQL, and configure the database interface properties. Start a database bridge (Oracle, Sybase, or ODBC), then click the Manual Connect and Login button in the database interface properties dialog. For more information, see the *G2 Database Bridge User's Guide*.

System Metrics

The System Metrics report builds a report from system performance counters such as memory utilization, CPU utilization, and number of events.

Note You must be Developer mode to configure the System Metrics report.

To configure system metrics, click the System Metrics tab, enable the Enable System Performance Metrics option, and configure the lag time, update interval, number of historical values, and historical age of the data.



☑ Enable System Performance Metrics

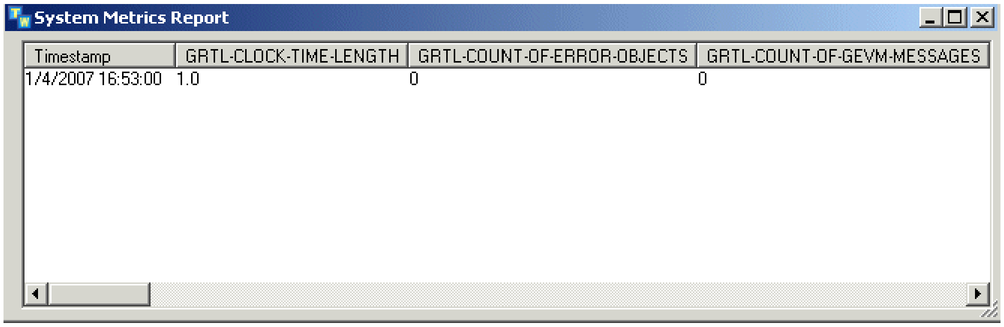
Lag Time: 60

Update Interval: 60

Historical Values: 120

Historical Age: 0

To show the domain objects report, in the Navigator, expand the Reports > System Settings node and choose Report to show the report. For example:

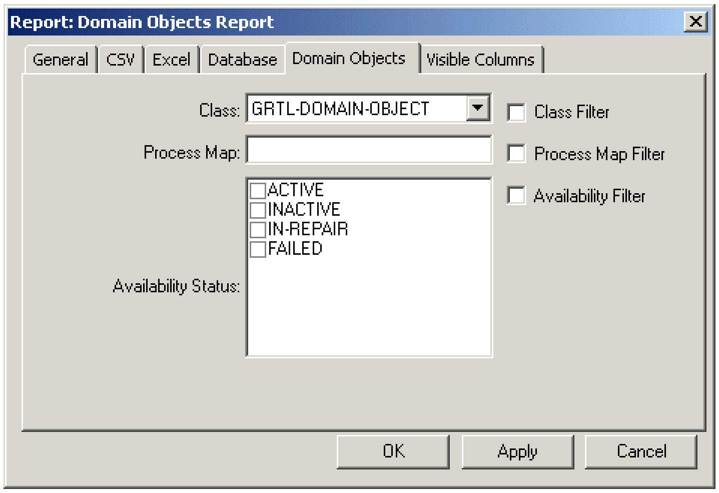


Timestamp	GRTL-CLOCK-TIME-LENGTH	GRTL-COUNT-OF-ERROR-OBJECTS	GRTL-COUNT-OF-GEVM-MESSAGES
1/4/2007 16:53:00	1.0	0	0

Domain Object Report

The Domain Object report collects values from domain objects (instances of grtl-domain-object). You can filter domain objects, based on the class, process map, and availability status, which you must specifically enable or disable in the dialog. You can also configure the attributes to show.

To configure the domain objects to show, click the Domain Objects tab:



Class: GRTL-DOMAIN-OBJECT

□ Class Filter

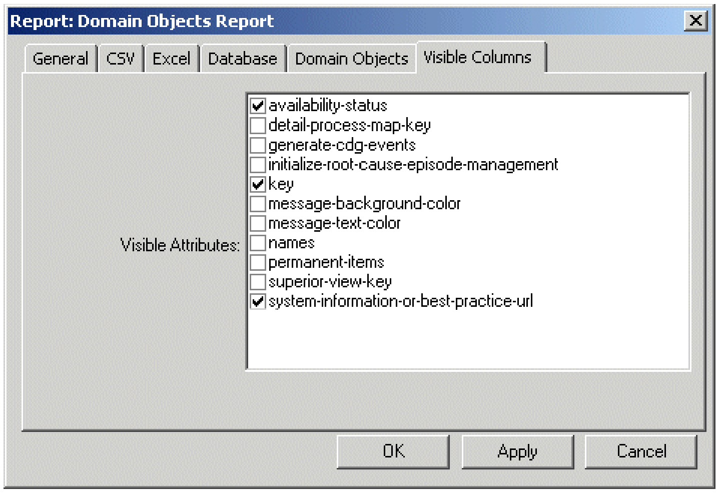
Process Map:

□ Process Map Filter

Availability Status:
□ACTIVE
□INACTIVE
□IN-REPAIR
□FAILED

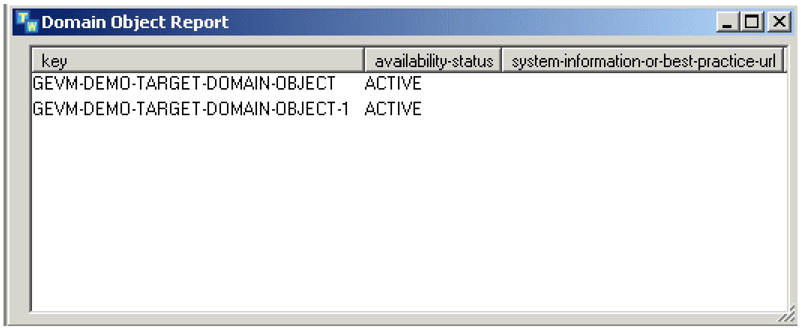
□ Availability Filter

To configure the attributes to show, click the Visible Columns tab:



Visible Attributes:
☑availability-status
□detail-process-map-key
□generate-cdg-events
□initialize-root-cause-episode-management
☑key
□message-background-color
□message-text-color
□names
□permanent-items
□superior-view-key
☑system-information-or-best-practice-url

To show the domain objects report, in the Navigator, expand the Reports > Domain Objects node and choose Report to show the report. For example:



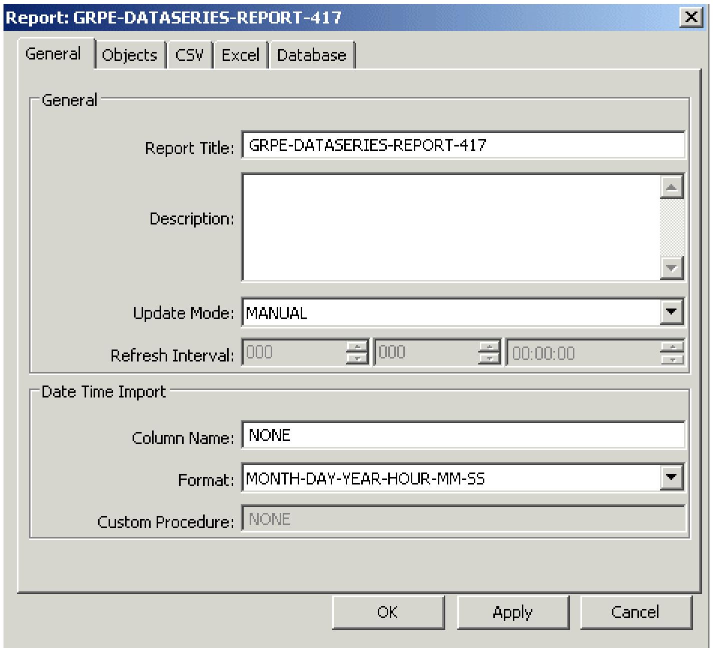
Domain Object Report

key	availability-status	system-information-or-best-practice-url
GEVM-DEMO-TARGET-DOMAIN-OBJECT	ACTIVE	
GEVM-DEMO-TARGET-DOMAIN-OBJECT-1	ACTIVE	

Data Series Report

The Data Series report is a generic report that enables you to easily import data series into G2 from objects, CSV files, or databases, to view and chart the data series, and to export the data to CSV files, Excel, or databases.

You can configure the date/time format to use on the General tab:



Report: GRPE-DATASERIES-REPORT-417

General

Report Title: GRPE-DATASERIES-REPORT-417

Description:

Update Mode: MANUAL

Refresh Interval: 000 000 00:00:00

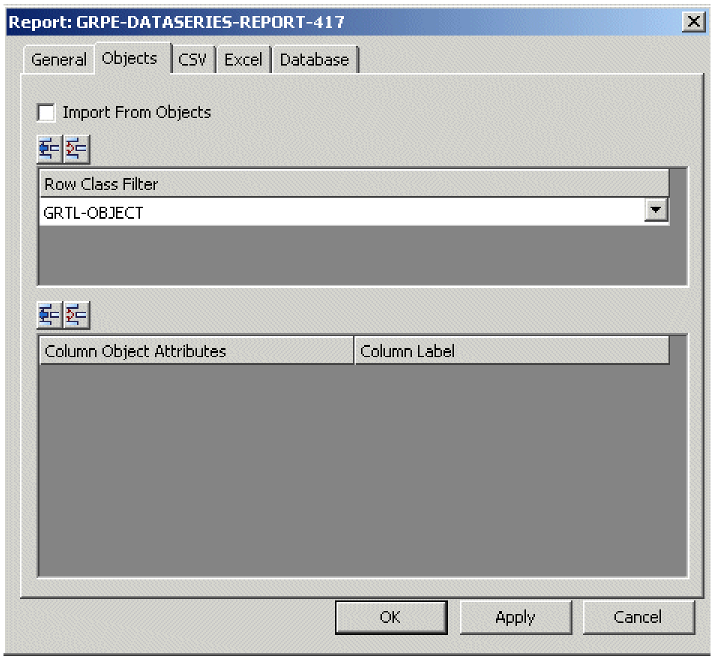
Date Time Import

Column Name: NONE

Format: MONTH-DAY-YEAR-HOUR-MM-SS

Custom Procedure: NONE

To import data from objects, click the Objects tab, enable the Import from Objects option, and configure the object class and attributes to import by adding rows and configuring the columns:

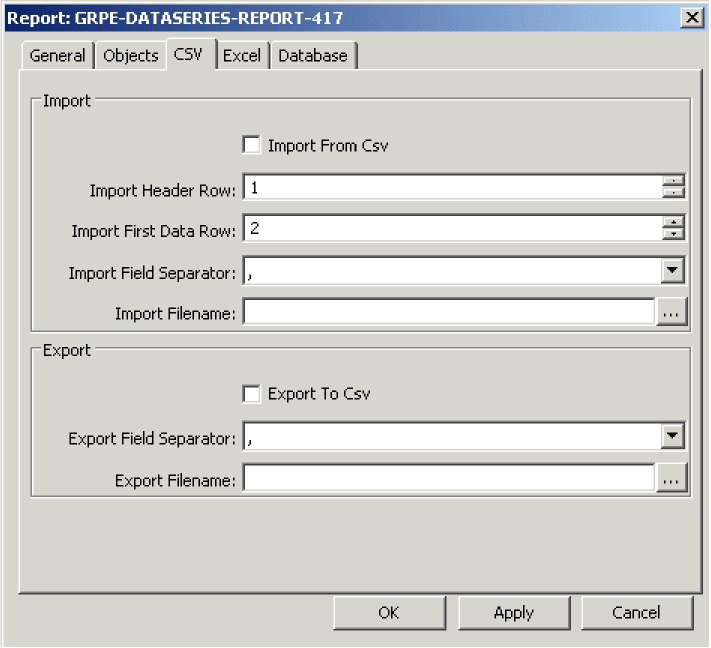


□ Import From Objects

Row Class Filter
GRTL-OBJECT

Column Object Attributes	Column Label

To import data from a CSV file, click CSV, enable the Import from CSV option, and configure the header row, first data row, separator, and filename:



Import

□ Import From Csv

Import Header Row: 1

Import First Data Row: 2

Import Field Separator: ,

Import Filename:

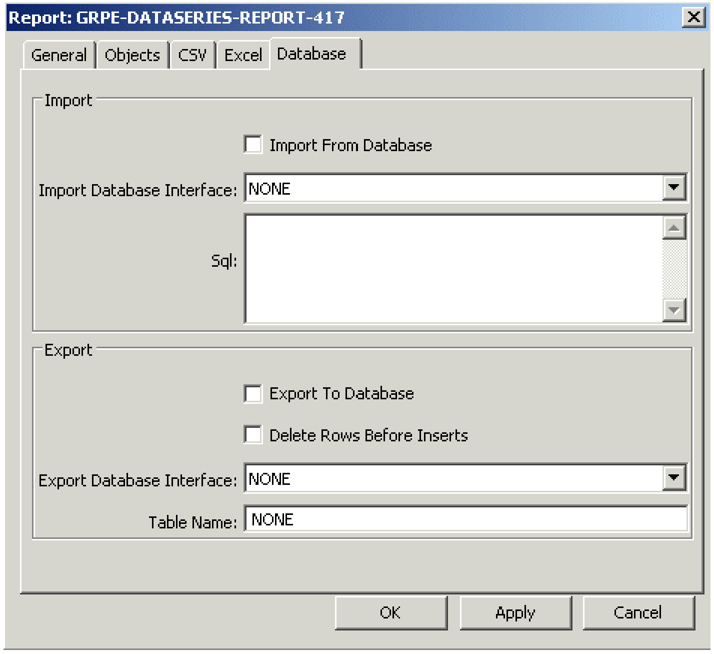
Export

□ Export To Csv

Export Field Separator: ,

Export Filename:

To import data from a database, click the Database tab, enable the Import from Database option, choose the database interface from which to import, and configure the SQL statement to use for importing the data:



Report: GRPE-DATASERIES-REPORT-417

Import

□ Import From Database

Import Database Interface: NONE

Sql:

Export

□ Export To Database

□ Delete Rows Before Inserts

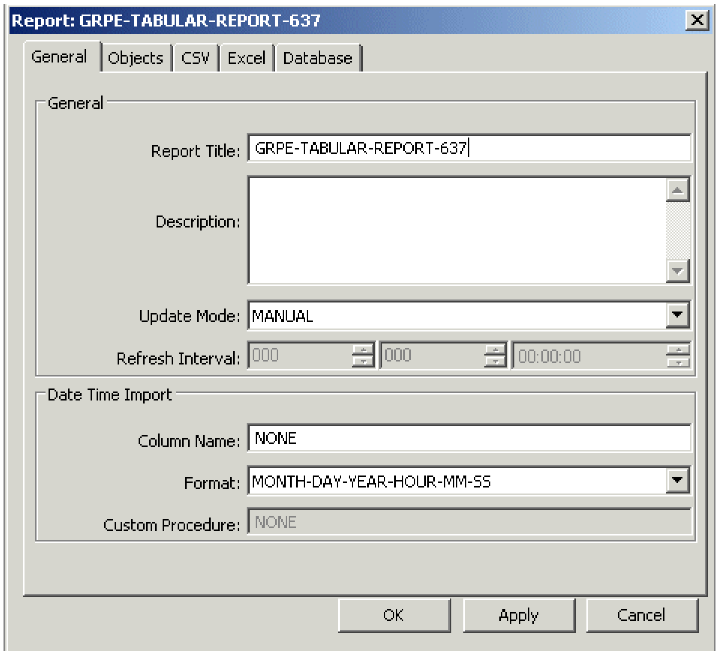
Export Database Interface: NONE

Table Name: NONE

Tabular Report

The Tabular report is similar to the Data Series report, except that it enables you to import and export all kinds of tabular values, not just data series. The content of Tabular reports cannot be charted. The common tabs of the report configuration dialog contains additional attributes that you can configure.

You can configure the date/time format to use on the General tab:



Report: GRPE-TABULAR-REPORT-637

General

Report Title: GRPE-TABULAR-REPORT-637

Description:

Update Mode: MANUAL

Refresh Interval: 000 000 00:00:00

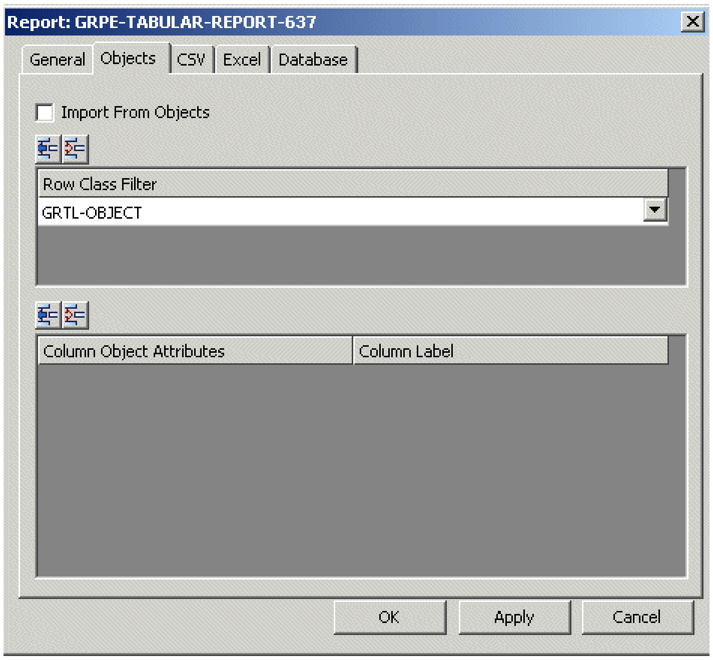
Date Time Import

Column Name: NONE

Format: MONTH-DAY-YEAR-HOUR-MM-SS

Custom Procedure: NONE

To import data from objects, click the Objects tab, enable the Import from Objects option, and configure the object class and attributes to import by adding rows and configuring the columns:

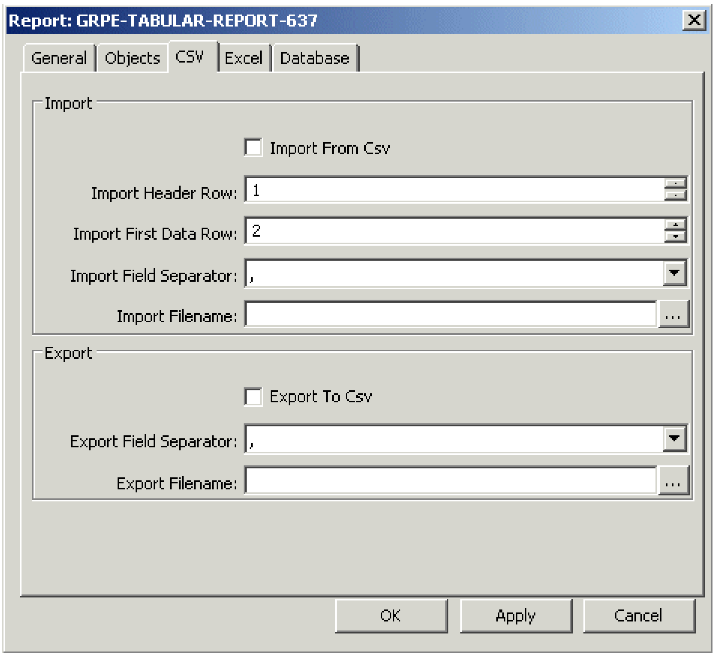


□ Import From Objects

Row Class Filter
GRTL-OBJECT

Column Object Attributes	Column Label

To import data from a CSV file, click CSV, enable the Import from CSV option, and configure the header row, first data row, separator, and filename:



Import

□ Import From Csv

Import Header Row: 1

Import First Data Row: 2

Import Field Separator: ,

Import Filename:

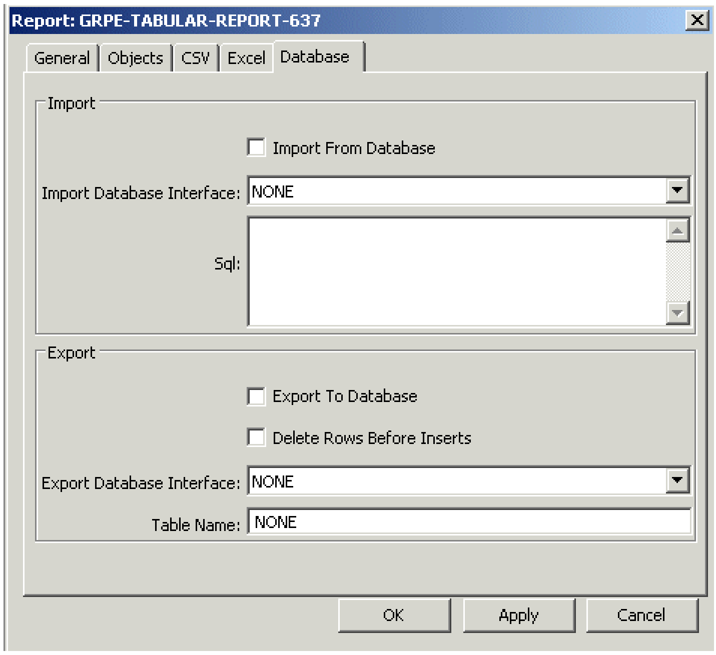
Export

□ Export To Csv

Export Field Separator: ,

Export Filename:

To import data from a database, click the Database tab, enable the Import from Database option, choose the database interface from which to import, and configure the SQL statement to use for importing the data:



Report: GRPE-TABULAR-REPORT-637

Import

□ Import From Database

Import Database Interface: NONE

Sql:

Export

□ Export To Database

□ Delete Rows Before Inserts

Export Database Interface: NONE

Table Name: NONE

Report Classes

Describes the built-in report classes in the G2 Reporting Engine (GRPE) module.

Introduction

This chapter describes the built-in classes and their related APIs in the GRPE module. The classes are:

- The base report class, grpe-report, upon which the other report classes are built.
- The grpe-realtime-report class, which is used for creating reports that update in real time.
- The grpe-tabular-report class, which is used for creating reports that consist of one or more tabular views of the data in which each tabular view's data is stored in a datatable.

- The grpe-dataseries-report, which is used for creating tabular reports that support charting the report data.
- The grpe-datatable class, which is used for storing data in tabular and dataseries reports.

Base Report Class

The base class for all reports is grpe-report, which is an abstract class.

Class Inheritance Path

grpe-report, object

Attributes

Attribute	Description
report-title	The report title.
Allowable values:	text
Default value:	""
description	A textual description for the report.
Allowable values:	text
Default value:	""

APIs

The following APIs are implemented by each of the subclasses of grpe-report to define the specific report behavior.

grpe-reset-report
(*report*: class grpe-report, *arg*: item-or-value)

Resets the report.

grpe-update-report
(*report*: class grpe-report, *arg*: item-or-value)

Updates the content of the report, that is, collects values.

grpe-show-report
(*report*: class grpe-report, *win*: class ui-client-item)

Opens a new view for the report.

grpe-update-views

(*report*: class grpe-report, *arg*: item-or-value)

Refreshes the content of all report views. This API is typically called when the values have been updated.

Real Time Reports

The grpe-realtime-report-pane class defines a tabular report that displays a dashboard of information in a pane. Real time reports are typically used to display and automatically update key information, such as performance indicators.

You define and configure instances of real time reports, which appear in the View menu. Real time reports are displayed in dockable panes, typically on the bottom of the screen.

Class Inheritance Path

grpe-realtime-report-pane, grpe-report, object

Attributes

Attribute	Description
dialog-id	A symbol for the dialog ID.
Allowable values:	symbol
Default value:	grpe-realtime-tabular-report-pane
class-filter	A grtl-object that is a filter for the report.
Allowable values:	symbol
Default value:	grtl-object
column-names	A sequence of symbols or texts for the column headers. When specified as symbols, the column headers are the text equivalent of the symbolic value, for example, my-header becomes “My Header”).
Allowable values:	sequence
Default value:	sequence()
_items	The items whose data to display in the report.

Attribute	Description
Allowable values:	sequence
Default value:	sequence()
_data	The data to display in the report.
Allowable values:	sequence
Default value:	sequence()

APIs

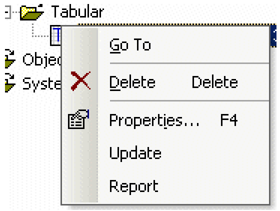
Typically, you create a subclass of grpe-realtime-report-pane and implement the following methods:

- grpe-reset-report: Typically configures the column-names attribute with the columns to display as a sequence of symbols, and configures the _items attribute to be the G2 items containing the values to display in the real time report. This API only works if column-names is specified as a sequence of symbols.
- grpe-update-report: Typically collects the values for the specified columns from the column-names attribute and _items attribute, and stores the values to display in the report in the _data attribute as a sequence of rows, where each row is a sequence of cell values.

Tabular Report

The grpe-tabular-report class defines a report that contains more than one tabular view forming the data set. The Tabular report stores the report data in a datatable, which is an instance of grpe-datatable. See “Datatable” on page 24.

The Tabular report defines the following popup menu:



Class Inheritance Path

grpe-tabular-report, grpe-report, object

Attributes

Attribute	Description
update-mode	How to update the report.
Allowable values:	manual, clock-time
Default value:	manual
update-interval	When update-mode is clock-time, the interval for updating the report.
Allowable values:	quantity
Default value:	0
_update-timeout	The timeout for updating the report.
Allowable values:	an instance of a _grpe-timeout
Default value:	an instance of a _grpe-timeout

Attribute	Description
datatable	Used for creating reports that contain more than one tabular view forming the data set.
Allowable values:	an instance of a grpe-datatable
Default value:	an instance of a grpe-datatable

APIs

grpe-reset-report
(*report*: class grpe-tabular-report, *arg*: item-or-value)

Initializes the datatable content.

grpe-update-views
(*report*: class grpe-tabular-report, *arg*: item-or-value)

Exports the datatable content.

grpe-update-report
(*report*: class grpe-tabular-report, *arg*: item-or-value)

Imports the datatable content.

grpe-show-report
(*report*: class grpe-tabular-report, *win*: class ui-client-item)

Displays the content of the report in a new view. Currently, this method is only supported in Telewindows Next Generation.

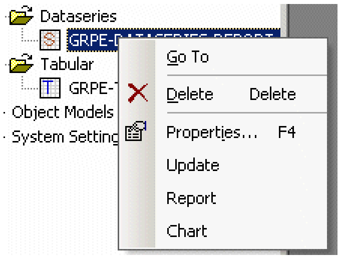
grpe-get-datatable
(*report*: class grpe-tabular-report, *index-or-name*: value)
–> *data-table*: class grpe-datatable

Returns a datatable, where *index-or-name* is the name of the data table or the index, which is only used if the report uses multiple data tables.

Data Series Report

The grpe-dataseries-report class is a subclass of grpe-tabular-report that supports charting the data.

The Data Series report defines the following popup menu:



Class Inheritance Path

grpe-dataseries-report, grpe-report, object

Attributes

Attribute	Description
chart	The chart associated with the report.
Allowable values:	An instance of a grpe-dataseries-chart
Default value:	An instance of a grpe-dataseries-chart

Datatable

The grpe-datatable class is used to store the configuration of tabular reports and collected values.

Class Inheritance Path

grpe-datatable-report, grtl-object-with-key

Attributes

Attribute	Description
_initialized	Whether the datatable is initialized.
Allowable values:	truth-value
Default value:	false
_items	The items in the datatable.
Allowable values:	sequence
Default value:	sequence()
_data	The data values in the datatable.
Allowable values:	sequence
Default value:	sequence()
column-specifications	The column headers in the datatable.
Allowable values:	sequence
Default value:	sequence()
dialog-id	The dialog ID as a symbol.
Allowable values:	symbol

Attribute	Description
Default value:	grpe-tabular-datatable-view
import-last-update-time	The time at which the datatable was last updated.
Allowable values:	quantity
Default value:	0
import-datetime-column-name	The column name that contains the date/time information.
Allowable values:	symbol
Default value:	none
import-datetime-format	The format of the date/time information in the column specified in the import-datetime-column-name attribute.
Allowable values:	year-month-day-hour-mm-ss, month-day-year-hour-mm-ss, day-month-year-hour-mm-ss, hour-mm-ss, hour-mm@:hh.hh, hh.hh, dd.dd, g2-time-stamp
Default value:	month-day-year-hour-mm-ss
import-date-custom-procedure	A procedure to convert date/time if custom formats are used.
Allowable values:	symbol
Default value:	none
import-csv-enabled	If true, imports values from a CSV file.
Allowable values:	truth-value
Default value:	true

Attribute	Description
import-csv-last-filename	The name of a CSV file from which to import values.
Allowable values:	text
Default value:	""
import-csv-header-row	The row number that contains the header information.
Allowable values:	integer
Default value:	1
import-csv-first-data-row	The first row that contains tabular values.
Allowable values:	integer
Default value:	2
import-csv-field-separator	The separator between values in each row when importing from a CSV file.
Allowable values:	text
Default value:	","
import-database-enabled	If true, imports values from a database.
Allowable values:	truth-value
Default value:	false
import-database-interface	The gdsm-database-interface to use when importing values from a database.
Allowable values:	symbol

Attribute	Description
Default value:	none
import-database-sql	The SQL statement for importing values from a database.
Allowable values:	text
Default value:	""
import-objects-enabled	If true, imports values from G2 objects.
Allowable values:	truth-value
Default value:	false
import-objects-class-filter	A sequence of class names, as symbols, from which to import values when importing from G2 objects.
Allowable values:	sequence
Default value:	sequence(the symbol grtl-object)
export-csv-enabled	If true, exports values to a CSV file.
Allowable values:	truth-value
Default value:	false
export-csv-filename	The name of the CSV file to which to export values.
Allowable values:	text
Default value:	""

Attribute	Description
export-csv-field-separator	The separator between values in each row when exporting to a CSV file.
Allowable values:	text
Default value:	","
export-excel-enabled	If true, exports values to all Excel processes connected to G2 via G2 ActiveXLink.
Allowable values:	truth-value
Default value:	false
export-excel-interface-class-name	The class name of the gsi-interface to use for optionally filtering the connected Excel processes when exporting data to Excel.
Allowable values:	symbol
Default value:	g2com-interface
export-excel-sheet-name	The name of the Excel sheet in which to export values.
Allowable values:	text
Default value:	""
export-excel-time-unit	The time unit of the values displayed in Excel. Time unit values will be scaled based on the value of this attribute.
Allowable values:	quantity
Default value:	1 second
export-excel-format-sheet-macro	An Excel macro to run for formatting the sheet containing the exported values.

Attribute	Description
Allowable values:	text
Default value:	"DefaultFormatSheet"
export-excel-format-data-macro	An Excel macro to run for formatting the exported values.
Allowable values:	text
Default value:	"DefaultFormatData"
export-excel-custom-row-data	A sequence of texts, each of which describes the contents of a cell displayed in the second row in the Excel sheet for that report. This attribute enables report subclasses to specify custom fields for the second row without having to regenerate the values each time the sheet updates.
Allowable values:	sequence
Default value:	sequence()
export-database-enabled	If true, exports values to a database.
Allowable values:	truth-value
Default value:	false
export-database-delete-rows-before-inserts	If true, deletes all rows in the database before inserting new ones.
Allowable values:	truth-value
Default value:	false
export-database-interface	The gdsm-database-interface to use when exporting values to a database.
Allowable values:	symbol

Attribute	Description
Default value:	none
export-database-table-name	The database table name in which to export values.
Allowable values:	symbol
Default value:	none

APIs

grpe-reset
(*datatable*: class grpe-datatable, *arg*: item-or-value)

Resets the content of the report. If the report imports values from G2 objects, collects the instances from which values will be collected.

grpe-import
(*datatable*: class grpe-datatable, *arg*: item-or-value)

Imports values from input source(s). Note that if multiple input sources are defined, the content is not guaranteed as one import will overwrite another one.

grpe-export
(*datatable*: class grpe-datatable, *arg*: item-or-value)

Exports the values to all selected output sources, including CSV files, Excel workbooks, databases, and open Telewindows Next Generation views. Note that Excel workbooks are only updated if Excel opened a live connection to G2. To do so, Excel needs to include the G2 add-in module and connect to G2. For details, see Chapter 5, “Excel Integration” on page 57.

grpe-get-value
(*datatable*: class grpe-datatable, *column*: integer, *row:* integer)
–> *value*: value

Gets the value of a cell given the *column* and *row*.

grpe-get-dataseries
(*datatable*: class grpe-datatable, *column*: integer)
–> *dataseries*: sequence

Gets the values defined in a *column*, typically a dataseries.

grpe-get-number-of-rows
(*datatable*: class grpe-datatable)
–> *rows*: integer

Return the number of rows.

grpe-get-number-of-columns
(*datatable*: class grpe-datatable)
–> *columns*: integer

Return the number of columns.

Charting

Describes the charting capabilities of the GRPE module.

Introduction

GRPE provides a layer on top of the built-in G2 charting package, based on the ProEssentials™ package from Gigasoft, Inc. (`http://www.gigasoft.com`). GRPE speeds up and simplifies chart creation by providing these objects:

- grpe-chart — A chart object, which represents the data to be displayed in the chart. You use data setter methods to set the data in a chart.
- grpe-chart-view — A chart view object, which describes how the chart should appear when displayed in Telewindows Next Generation. The chart that appears in Telewindows Next Generation is referred to as a *native chart*.

A chart can have any number of views, which you create and delete by using menu choices on the chart object. If you show a chart that defines no views, GRPE automatically creates a chart view object. GRPE creates a native chart from a chart and a chart view. A view can have only one native chart showing per window.

For examples of configuring charts, see `grpe-demo.kb` located in the `g2i\examples` directory.

Creating and Configuring Charts

To create and configure a chart, you create a chart object, and create and configure one or more chart views.

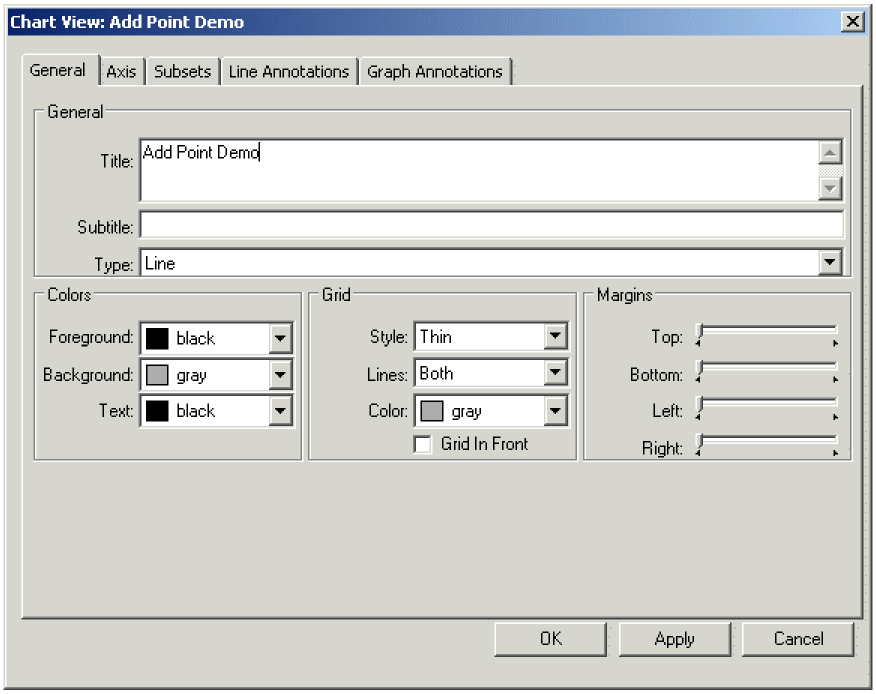
Configuring Chart Objects

A grpe-chart object defines three menu choices:

- Show Chart — Shows a native chart in a window. If the chart defines multiple chart views, a submenu appears for choosing the view you want to show.
- Create View — Creates a view for a chart, which you can configure. A chart can have one or more chart views.
- Delete View — Deletes a view for a chart. This menu choice only appears if the chart has a chart view. If the chart defines multiple chart views, a submenu appears for choosing the view you want to delete.

Configuring Chart Views

Here is the General tab, which allows you to configure the chart title and subtitle, chart type, colors, grid lines, and margins:



General

Title: Add Point Demo

Subtitle:

Type: Line

Colors

Foreground: black

Background: gray

Text: black

Grid

Style: Thin

Lines: Both

Color: gray

☐ Grid In Front

Margins

Top:

Bottom:

Left:

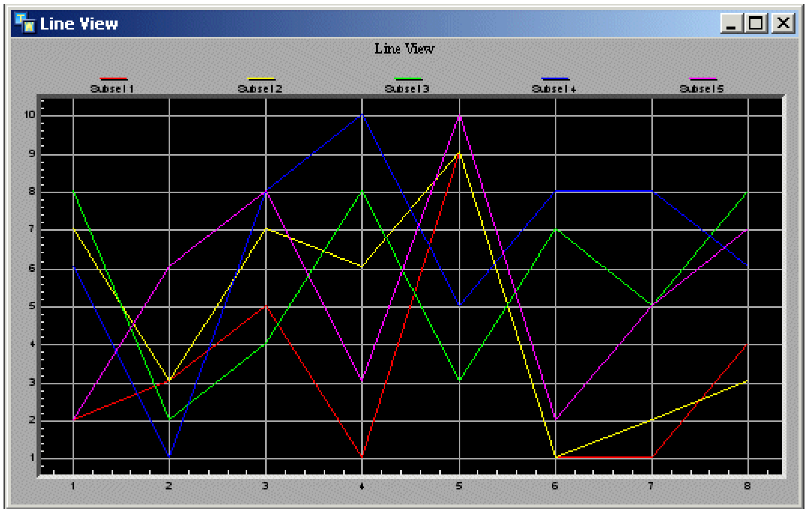
Right:

GRPE allows you to provide data in any number of formats, and it uses the data it has available to display the particular type of chart you select in the best way it can. For example, if you do not provide Z data and you choose a 3D scatter chart, the chart looks like a 2D scatter chart with the Z coordinate of every point equal to 0.

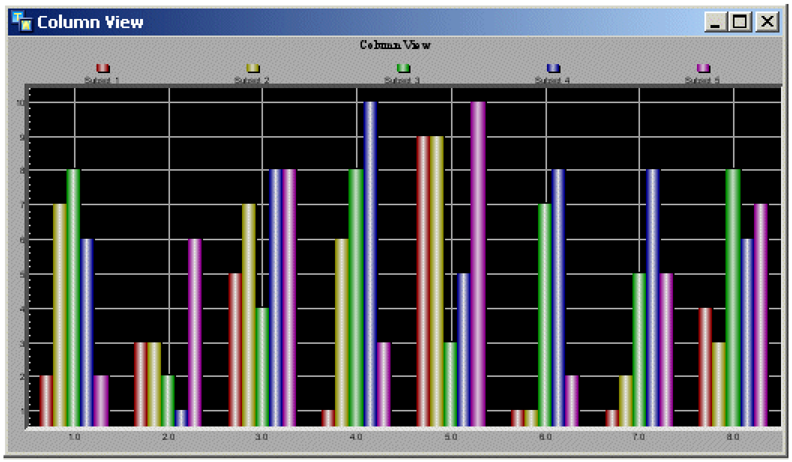
Chart Types

Here are examples of each of the types of charts that you can create, using multiple data subsets:

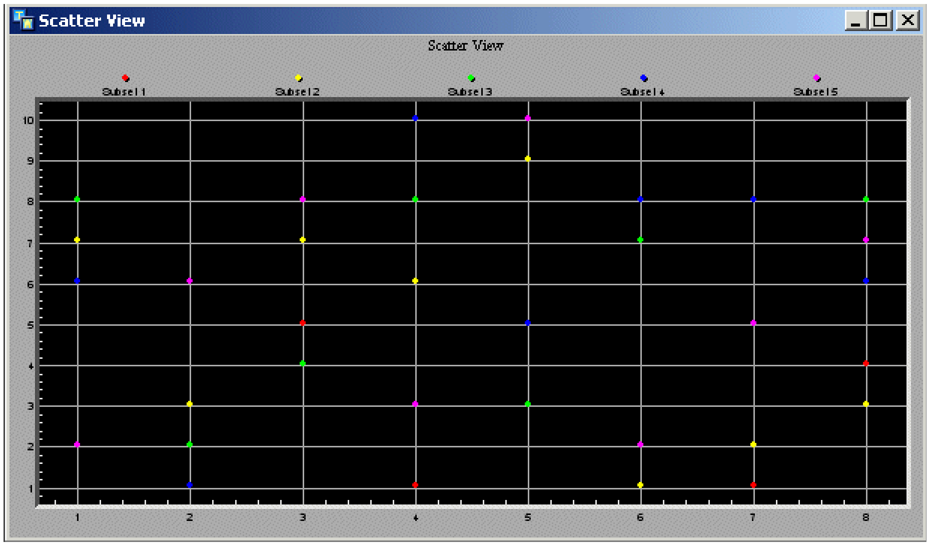
- Line



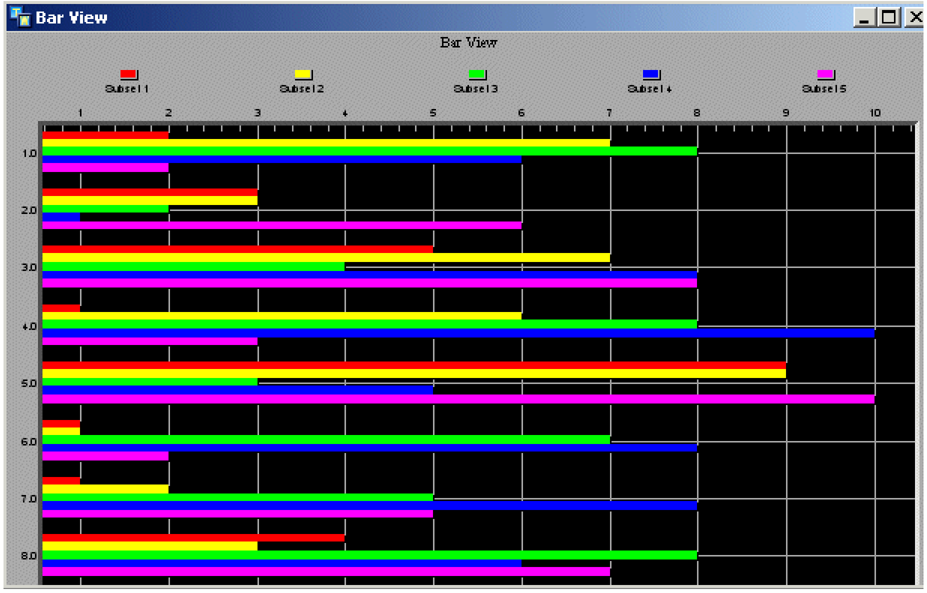
- Column



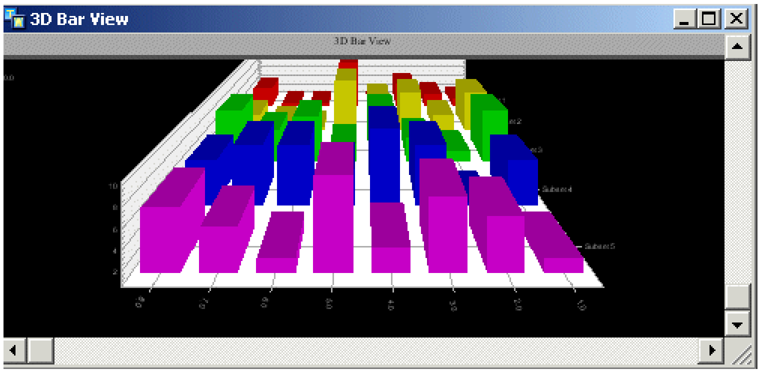
- Scatter


Scatter View
Scatter View
Subset 1
Subset 2
Subset 3
Subset 4
Subset 5
10
9
8
7
6
5
4
3
2
1
1
2
3
4
5
6
7
8

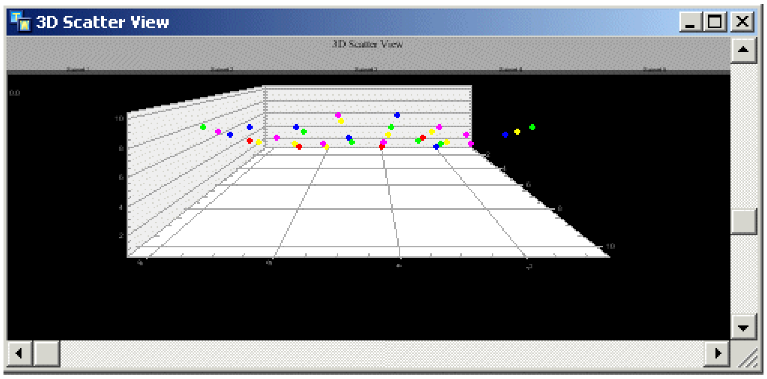
- Bar


Bar View
Bar View
Subset 1
Subset 2
Subset 3
Subset 4
Subset 5
1
2
3
4
5
6
7
8
9
10
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0

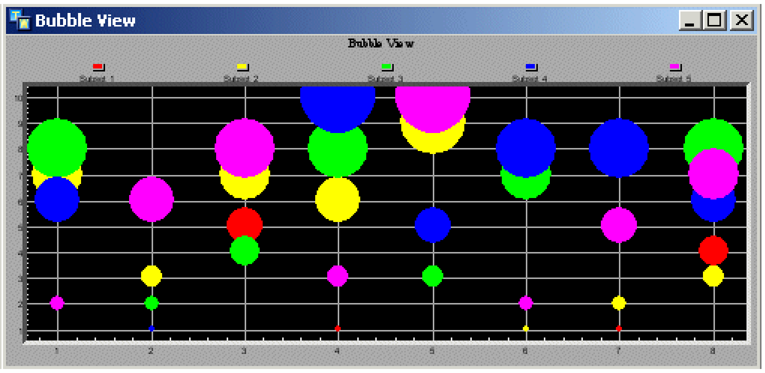
- 3D Bar



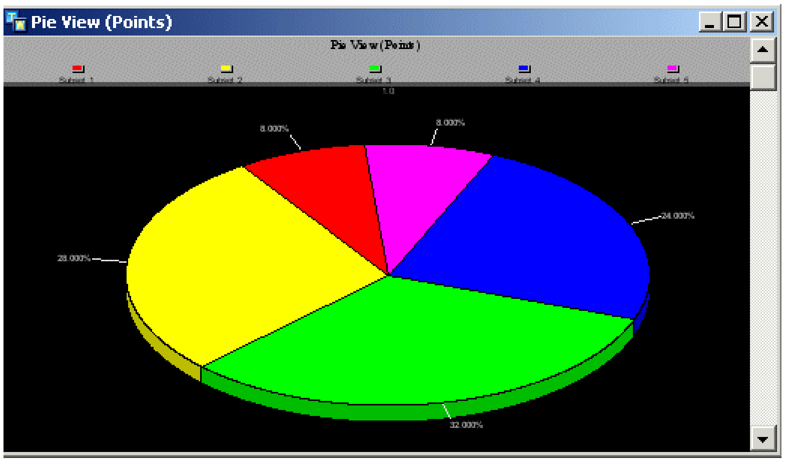
- 3D Scatter



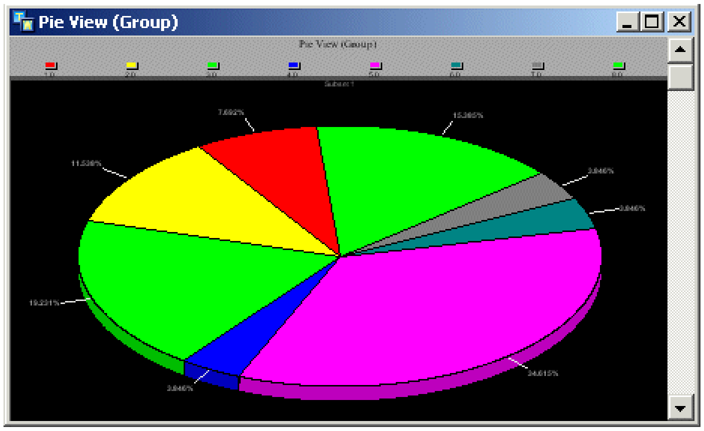
- Bubble



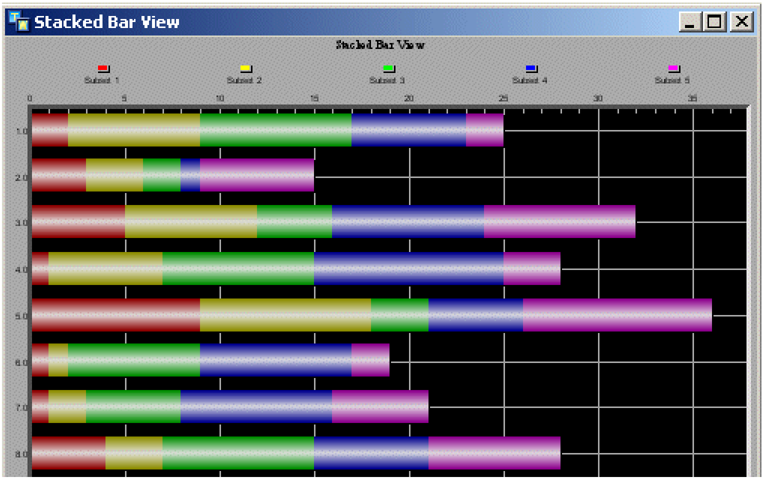
- Pie (Points) — A pie chart for each point where each wedge represents a group’s contribution to that point.



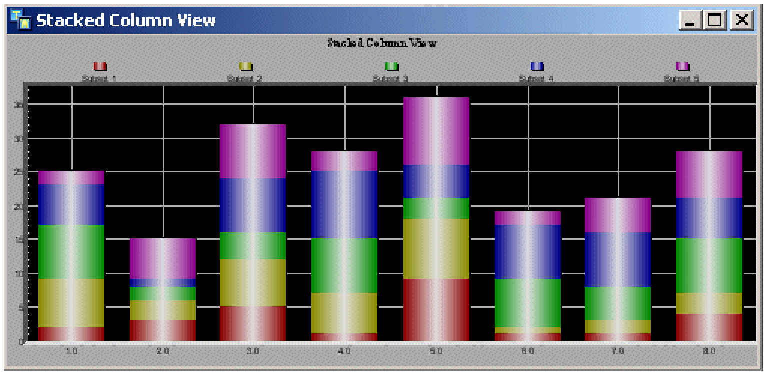
- Pie (Groups) — A pie chart for each group where each wedge represents the point’s contribution to that group.



- Stacked Bar

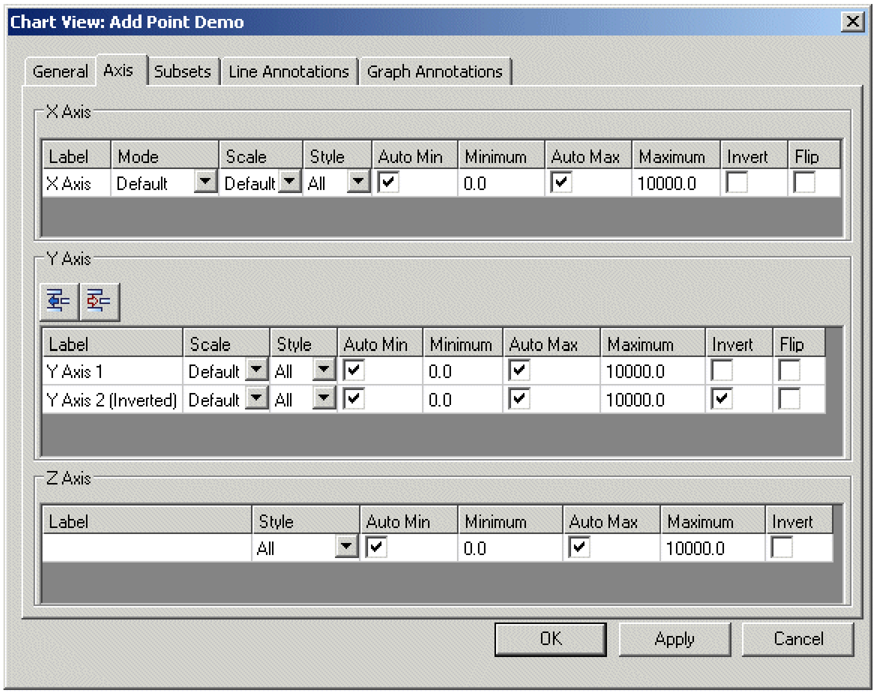


- Stacked Column



Axis Tab

Here is the Axis tab, which allows you to configure the X, Y, and Z axes of the chart:



X Axis

Label	Mode	Scale	Style	Auto Min	Minimum	Auto Max	Maximum	Invert	Flip
X Axis	Default	Default	All	☑	0.0	☑	10000.0	□	□

Y Axis

Label	Scale	Style	Auto Min	Minimum	Auto Max	Maximum	Invert	Flip
Y Axis 1	Default	All	☑	0.0	☑	10000.0	□	□
Y Axis 2 (Inverted)	Default	All	☑	0.0	☑	10000.0	☑	□

Z Axis

Label	Style	Auto Min	Minimum	Auto Max	Maximum	Invert
	All	☑	0.0	☑	10000.0	□

You configure the X and Y axes for all chart types. You can configure up to 16 Y axes, although you must have at least one. You configure the Z axis for any 3D chart.

For each axis, you configure the Label and Style, whose options are All, Grid/Labels, Grid/Labels, Only Grid, Only Labels, or None. For the X and Y axes, you also configure the Scale, whose options are Default, Linear, and Log. For the X axis, you also configure the Mode, whose options are Default and Time.

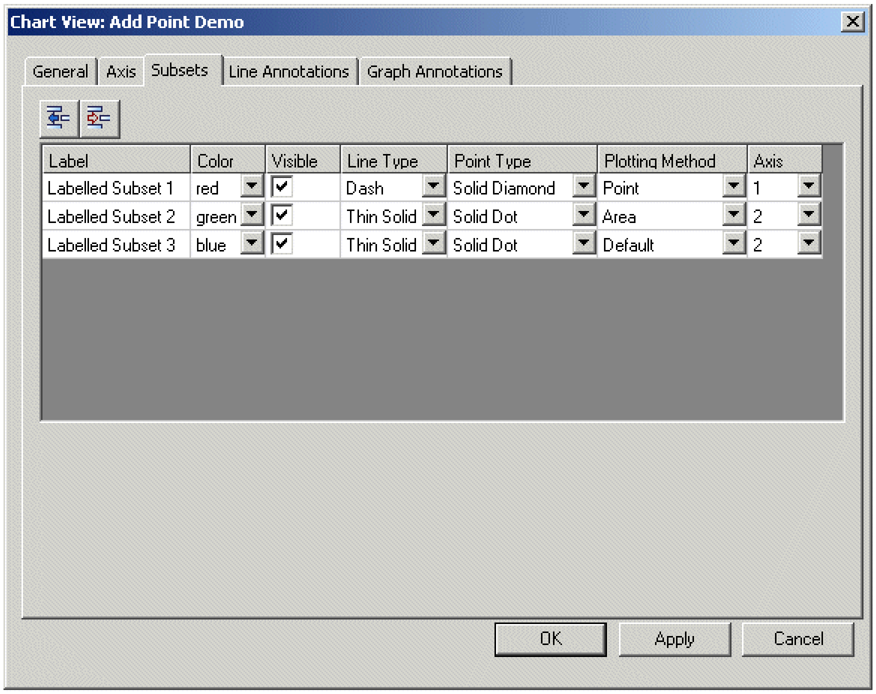
Enable the Auto Min and Auto Max check boxes to set the endpoints of the axis automatically, or disable Auto Min and Auto Max and configure the Minimum and Maximum values to manually set the endpoints.

Enable the Invert check box to invert the axis; for example, inverting the X axis causes the values to grow to the left rather than to the right.

Enable the Flip check box to shows the X axis on the top or the Y axis on the right instead of the default bottom/left positions.

Subsets Tab

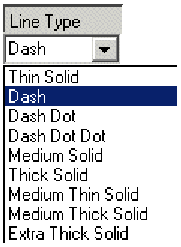
Here is the Subsets tab, which allows you to configure various properties of each subset:

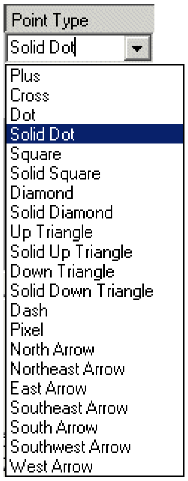


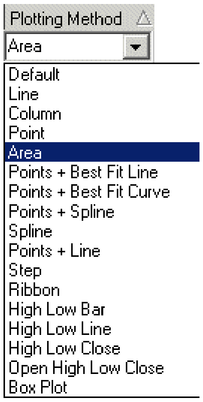
Label	Color	Visible	Line Type	Point Type	Plotting Method	Axis
Labelled Subset 1	red	☑	Dash	Solid Diamond	Point	1
Labelled Subset 2	green	☑	Thin Solid	Solid Dot	Area	2
Labelled Subset 3	blue	☑	Thin Solid	Solid Dot	Default	2

Label, Color, and Visible are relevant for all chart types. Line Type and Point Type are relevant for charts that have lines or points. Plotting Method is relevant for Line, Scatter, Column, and Bar charts, and allows you to mix different plotting methods in the same chart. Axis is relevant for Line, Scatter, and Column charts, and specifies the axis on which the subset should be plotted, where 1 refers to the first Y axis, 2 refers to the second Y axis, and so on.

To configure the Line Type, Point Type, and Plotting Method, click in a cell to display a dropdown list of options:



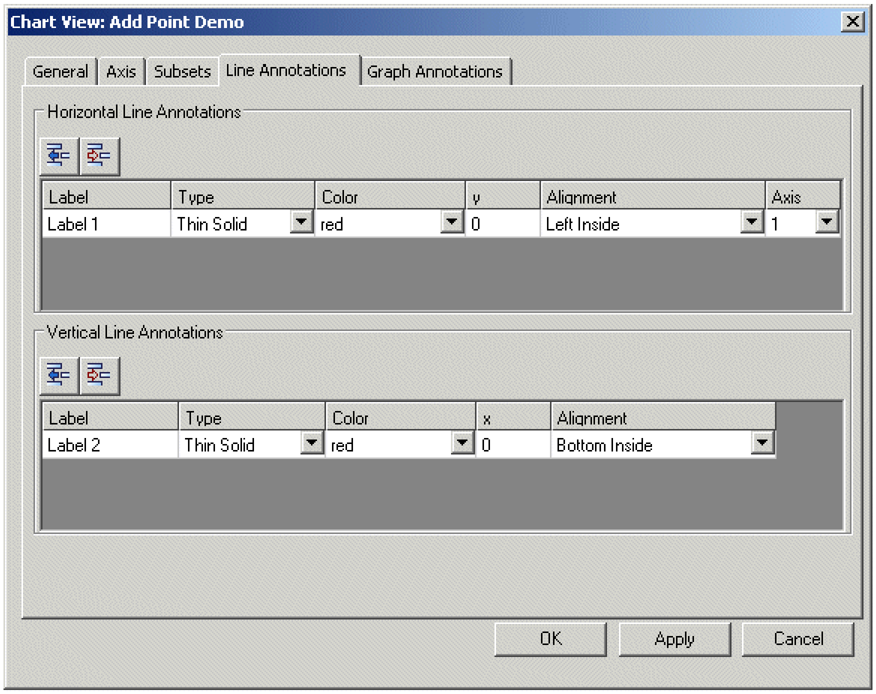




See Demo 2 in `grpe-demo.kb` for examples of using multiple axes and multiple plotting methods.

Line Annotations Tab

Here is the Line Annotations tab, which allows you to add horizontal and vertical line annotations to the chart:



Horizontal Line Annotations

Label	Type	Color	y	Alignment	Axis
Label 1	Thin Solid	red	0	Left Inside	1

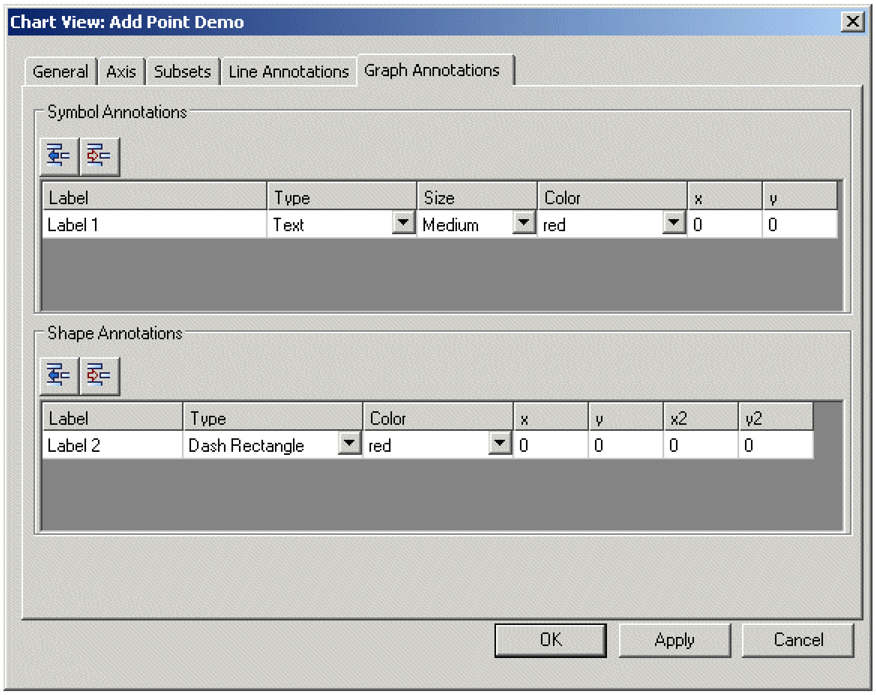
Vertical Line Annotations

Label	Type	Color	x	Alignment
Label 2	Thin Solid	red	0	Bottom Inside

For each annotation, you configure the Label, Type, Color, position of the line, and Alignment. For horizontal annotations, you also configure the Axis.

Graph Annotations Tab

Here is the Graph Annotations tab, which allows you to add symbols or shapes to a chart:



Symbol Annotations

Label	Type	Size	Color	x	y
Label 1	Text	Medium	red	0	0

Shape Annotations

Label	Type	Color	x	y	x2	y2
Label 2	Dash Rectangle	red	0	0	0	0

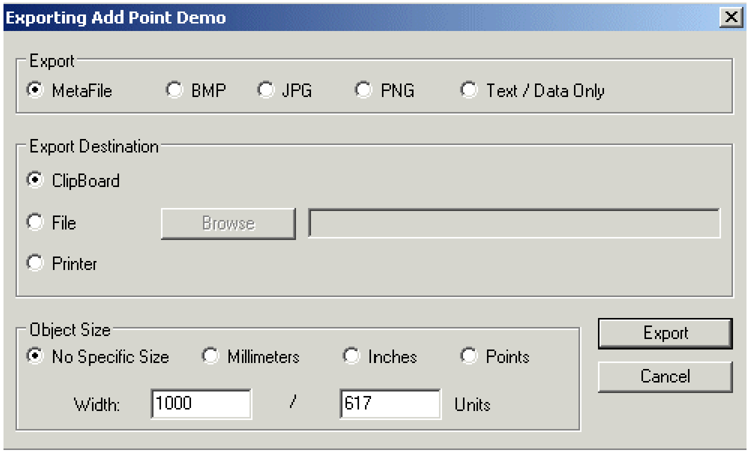
For each annotation, you configure the Label, Type, and Color. For symbol annotations, you also configure the Size. For symbols, you configure the position of the symbol, and for shapes, you configure the top-left and bottom-right coordinates.

Configuring Charts

In addition to the attributes you can modify on the property dialog, you can also modify some chart view attributes via the following menu choices on a displayed native chart:

- Properties — Displays the properties dialog for the associated chart object. See “Configuring Chart Views” on page 35.
- Update — Updates the native chart view using the current data.
- Font Size — Adjusts the font size of the chart view labels.
- Legend — Adjusts the location, spacing, and visibility of the color legend in the chart view. By default, it appears at the top of the view in two lines.

- Maximize — Expands the chart view to fit the entire screen. Click the title bar or press the Escape key to reduce it.
- Export Dialog — Exports the chart in one of several formats:



Exporting Add Point Demo

Export
● MetaFile ○ BMP ○ JPG ○ PNG ○ Text / Data Only

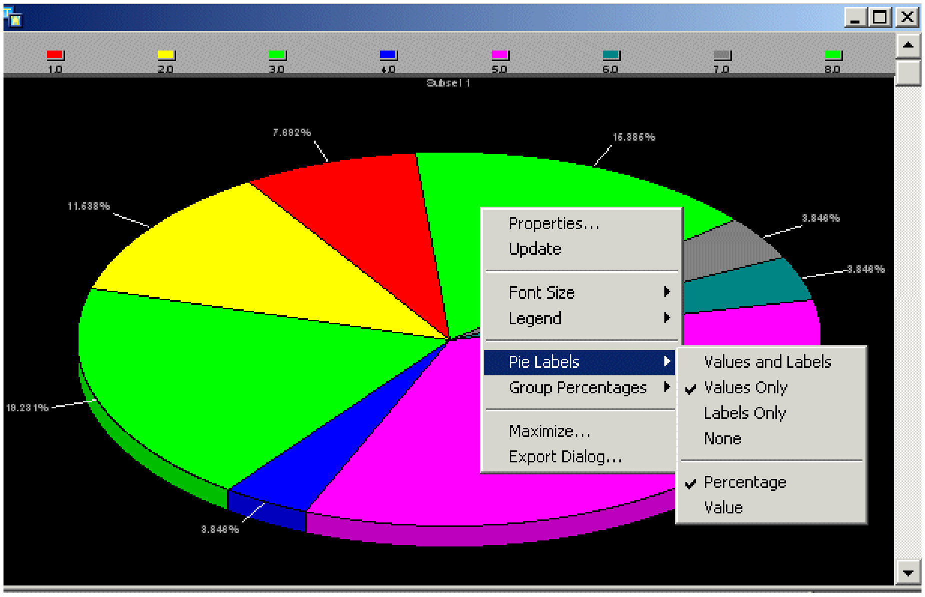
Export Destination
● ClipBoard
○ File Browse
○ Printer

Object Size
● No Specific Size ○ Millimeters ○ Inches ○ Points

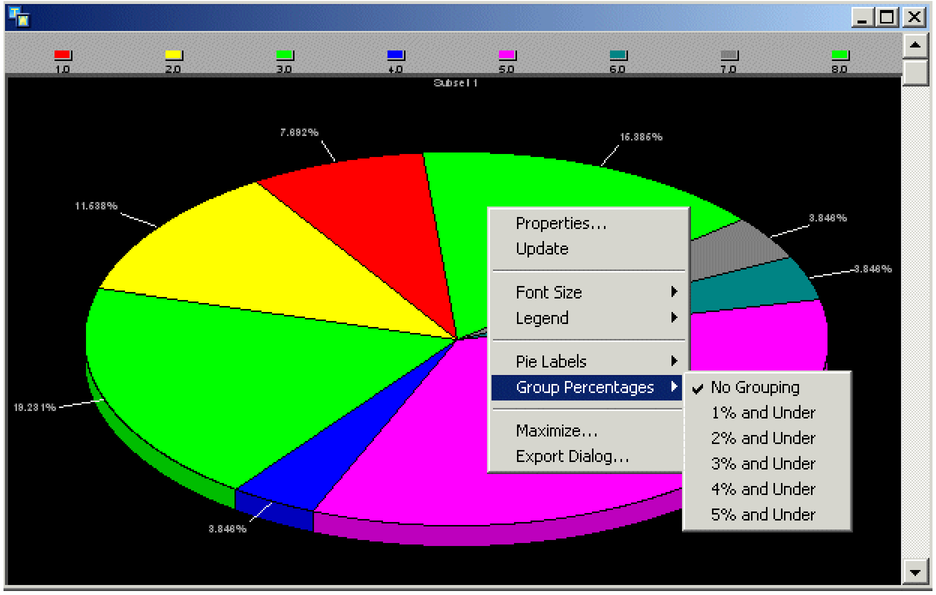
Width: 1000 / 617 Units

In addition, the two Pie charts contain these menu choices:

- Pie Labels — Collapses all wedges smaller than the specified amount into one wedge labelled Other:

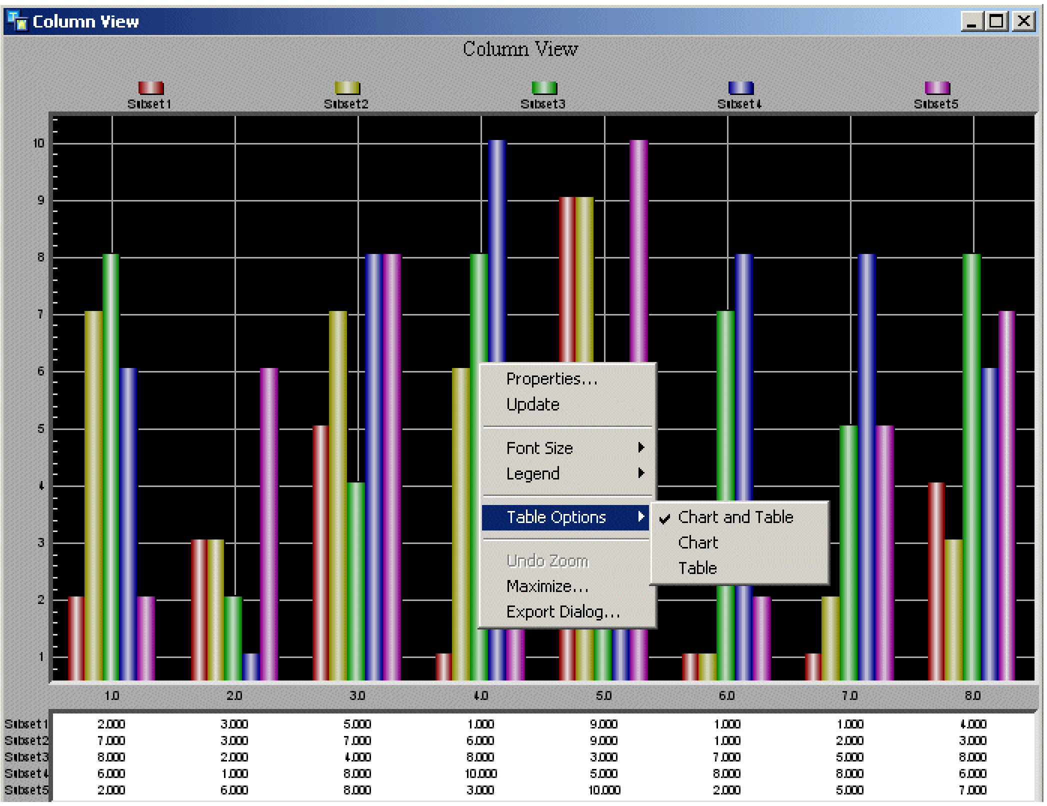


- Group Percentages — Adjusts the percentages shown for each slice in the pie:



The Column and Stacked Column charts contain this menu choice:

- Table Options — Displays the chart alone, the table alone, or both the chart and table:



	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0
Subset1	2.000	3.000	5.000	1.000	9.000	1.000	1.000	4.000
Subset2	7.000	3.000	7.000	6.000	9.000	1.000	2.000	3.000
Subset3	8.000	2.000	4.000	8.000	3.000	7.000	5.000	8.000
Subset4	6.000	1.000	8.000	10.000	5.000	8.000	8.000	6.000
Subset5	2.000	6.000	8.000	3.000	10.000	2.000	5.000	7.000

API

This section describes the APIs for creating and manipulating charts and chart views.

Chart

You use the following methods to show and update chart views for a given chart.

grpe-show-chart
(*chart*: class grpe-chart, *chart-view*: class grpe-chart-view,
win: class g2-window)

Shows the specified *chart-view* of the given *chart* in *win*.

grpe-show-chart
(*chart*: class grpe-chart, *win*: class g2-window)

Shows the default view of the specified *chart* in *win*.

grpe-hide-chart
(*chart*: class grpe-chart)

Hides the default view for *chart* in all windows.

grpe-hide-chart
(*chart*: class grpe-chart, *win*: class g2-window)

Hides the default view for *chart* in *win*.

grpe-hide-chart
(*chart*: class grpe-chart, *chart-view*: class grpe-chart-view,
win: class g2-window)

Hides the specified *chart-view* of the given *chart* in *win*.

grpe-update-chart
(*chart*: class grpe-chart, *chart-view*: class grpe-chart-view,
win: class g2-window)

Updates the specified *chart-view* of the given *chart* in *win*. If the view is not already displayed, this procedure does nothing.

grpe-update-chart
(*chart*: class grpe-chart, *win*: class g2-window)

Updates the default view of the specified *chart* in *win*.

grpe-update-chart
(*chart*: class grpe-chart)

Updates all views of the specified *chart* currently displayed on all windows.

Data Setters

You use data setters to specify the data to plot. Depending on the type of data, you use different methods to specify the data to plot, as follows:

To plot...	Use one of these data setters...
One point in a subset	grpe-add-point
A set of points that belong to the same group, subset, or curve	grpe-add-group grpe-set-group
A sequence of points, each of which should be added to a group present in the chart, where one point is added to each group.	grpe-add-points

GRPE provides different versions of each data setter with different arguments, depending on whether the plot contains X, Y, and/or Z data.

For example, Demo 1 uses grpe-set-group and repeatedly changes the Y data for five different groups, but never explicitly specifies the X data or Z data. Demo 2 uses grpe-add-points and grpe-add-point to incrementally add points to each of three groups.

In the data setter APIs that follow, the *subset-index-or-id* argument can be:

- A non-negative integer, which is treated as an index into the group, where the first group added is index 0, the second group added is index 1, and so on.
- A symbol or a text, in which case the method checks to see if a data setter has previously been called with this symbol or text as the *subset-index-or-id*, and if so, modifies that group. The first call to the *subset-index-or-id* after calling grpe-initialize-chart creates a new group with that ID, which can be used in future data setter calls.

grpe-initialize-chart
(*chart*: class grpe-chart)

Clears all the data in *chart*. You typically call grpe-initialize-chart before calling any of the data setters.

grpe-add-point
(*chart*: class grpe-chart, *subset-index-or-id*: value, *x*: float, *y*: float, *z*: float)

Adds a point with the specified *x*, *y*, and *z* values to the group specified by *subset-index-or-id* in the given *chart*.

grpe-add-point
(*chart*: class grpe-chart, *subset-index-or-id*: value, *x*: float, *y*: float)

Adds a point with the specified *x* and *y* values to the group specified by *subset-index-or-id* in the given *chart*.

grpe-add-points
(*chart*: class grpe-chart, *xdata*: sequence, *ydata*: sequence, *zdata*: sequence)

Adds one point to each group present in *chart*, where the number of elements in *ydata* must equal the number of groups in the chart. The *xdata* and *zdata* arguments must contain exactly one element or be the same size as *ydata*. If *xdata* or *zdata* have only one element, that value is used as the x or z value of all added points, with the y coordinate specified by *ydata*.

grpe-add-points
(*chart*: class grpe-chart, *xdata*: sequence, *ydata*: sequence)

Adds one point to each group present in *chart*, where the number of elements in *ydata* must equal the number of groups in the chart. The *xdata* argument must contain exactly one element or be the same size as *ydata*. If *xdata* only has one element, that value is used as the x value of all added points, with the y coordinate specified by *ydata*.

grpe-set-group
(*chart*: class grpe-chart, *subset-index-or-id*: value, *xdata*: sequence,
ydata: sequence, *zdata*: sequence)

Sets the *xdata*, *ydata*, and *zdata* of the group given by *subset-index-or-id* in *chart*. The *xdata* and *zdata* arguments must have the same number of elements as *ydata* or be empty, in which case the method behaves as if no *xdata* or *zdata* were passed in. The number of points in the resulting group equals the number of elements in *ydata*.

grpe-set-group
(*chart*: class grpe-chart, *subset-index-or-id*: value, *xdata*: sequence,
ydata: sequence)

Sets the *xdata* and *ydata* of the group given by *subset-index-or-id* in chart. The *xdata* argument must have the same number of elements as *ydata* or must be empty, in which case the method behaves as if no *xdata* was passed in. The number of points in the resulting group equals the number of elements in *ydata*. The corresponding Z data is implicit.

grpe-set-group
(*chart*: class grpe-chart, *subset-index-or-id*: value, *ydata*: sequence)

Sets the *ydata* of the group given by *subset-index-or-id* in *chart*. The number of points in the resulting group equals the number of elements in *ydata*. The corresponding X data and Z data are implicit.

grpe-add-group
(*chart*: class grpe-chart, *xdata*: sequence, *ydata*: sequence, *zdata*: sequence)
–> *subset-index-or-id:* value

Adds *xdata*, *ydata*, and *zdata* to a new group in *chart* and returns the *subset-index-or-id* of the group. The *xdata* and *zdata* must have the same number of elements as *ydata* or must be an empty sequence, in which case the method behaves as if no *xdata zdata* were passed in. The number of points in the resulting group equals the number of elements in *ydata*.

grpe-add-group

(*chart*: class grpe-chart, *xdata*: sequence, *ydata*: sequence)
–> *subset-index-or-id:* value

Adds *xdata* and *ydata* to a new group in *chart* and returns the *subset-index-or-id* of the group. The *xdata* must have the same number of elements as *ydata* or must be an empty sequence, in which case the method behaves as if no *xdata* were passed in. The number of points in the resulting group equals the number of elements in *ydata*. The corresponding Z data is implicit.

grpe-add-group

(*chart*: class grpe-chart, *ydata*: sequence)
–> *subset-index-or-id:* value

Adds *ydata* to a new group in *chart* and returns the *subset-index-or-id* of the group. The number of points in the resulting group equals the number of elements in *ydata*. The corresponding X data and Z data are implicit.

grpe-remove-group

(*chart*: class grpe-chart, *subset-index-or-id*: value)

Removes the group given by *subset-index-or-id* from *chart*.

Chart Views

You implement the following methods to create and initialize chart views.

grpe-create-view

(*chart*: class grpe-chart)
–> *chart-view*: class grpe-chart-view

Implement this method when you want to set the attributes of a grpe-chart-view at view creation time, for example, when the user chooses the Create View menu choice. The method should set the attributes of the resulting chart view.

grpe-initialize-view-for-update

(*chart*: class grpe-chart, *view*: class grpe-chart-view)

Implement this method when you want to set the attributes of a grpe-chart-view when the chart is updated. For example, to always display a chart as a 3d-scatter chart, you would set the chart-type of the view to be 3d-scatter in this method. Note that this method only gets called if _initialize-before-update of the grpe-chart-view is true.

Gantt Charts

A gantt chart consists of a number of groups, or subsets, along the Y axis and time increasing along the X axis. Each group consists of a set of time bars, where each time bar can have a particular state. For example, the gantt chart might represent the current utilization of a resource, where the state is busy.

Configuring a gantt chart differs from configuring a normal chart in these ways:

- General tab:
  - You do not configure grid options.
  - You do not configure the Chart Type.
- Subsets/States tab:
  - In addition to subsets, you configure the Label and Color of the various states.
  - You configure only the Label of each subset.

For an example, see Demo 3 in `grpe-demo.kb`.

API

You use the following methods to set the data in a gantt chart.

grpe-gantt-set-group-state
(*chart*: class grpe-chart, *subset-index-or-id*: value, *state*: value,
t1: quantity, *t2*: quantity)

Sets the time bar values of the group given by *subset-index-or-id* of *chart*. The bar goes from *t1* to *t2* along the X axis, where *t2* must be greater than or equal to *t1*.

For example, to create a gantt chart that shows that person-1 is busy from 1 to 9:

```
grpe-gantt-set-group-state (chart, the symbol person-1, the symbol busy, 1, 9)
```

To show that you are busy from 4 to 8:

```
grpe-gantt-set-group-state(chart, the symbol person-2, the symbol busy, 4, 8)
```

grpe-gantt-set-group-state
(*chart*: class grpe-chart, *subset-index-or-id*: value, *state*: value,
t1: quantity, *t2:* quantity, *user-data*: structure)

The same as the previous method with the addition of the optional *user-data* argument, which is used in conjunction with the hotspot callback. You can pass user data associated with a particular time bar, which will be passed to the hotspot-callback-procedure when the user clicks that time bar. For details, see “grpe-chart” on page 55.

grpe-gantt-toggle-group-state
(*chart*: class grpe-gantt-chart, *subset-index-or-id*: value, *state*: value,
t: quantity)

Toggles the state of the group given by *subset-index-or-id* of *chart* at time *t*. The time specified for a particular group must be greater than or equal to the greatest time value already specified for the group.

For example, the following two calls toggle the state on and off to set the state to busy between 1 and 9:

```
// toggle state on
call grpe-gantt-toggle-group-state(chart, the symbol person-1,
    the symbol busy, 1);

// toggle state off
call grpe-gantt-toggle-group-state(chart, the symbol person-1,
    the symbol busy, 9);
```

If the value of *state* differs when toggling the state off, the system procedure toggles the existing state off, then toggles the new state on. For example, the following two calls:

```
//toggle busy state on
call grpe-gantt-toggle-group-state(chart, the symbol person-1,
    the symbol busy, 1);

//toggle asleep state on, which implicitly toggles busy state off
call grpe-gantt-toggle-group-state(chart, the symbol person-1,
    the symbol asleep, 9);
```

are equivalent to:

```
//toggle busy state on
call grpe-gantt-toggle-group-state(chart, the symbol person-1,
    the symbol busy, 1);

//toggle busy state off
call grpe-gantt-toggle-group-state(chart, the symbol person-1,
    the symbol busy, 9);

//toggle asleep state on
call grpe-gantt-toggle-group-state(chart, the symbol person-1,
    the symbol asleep, 9);
```

grpe-gantt-toggle-group-state
(*chart*: class grpe-gantt-chart, *subset-index-or-id*: value, *state*: value,
t: quantity, *user-data*: structure)

The same as the previous method with the addition of the optional *user-data* argument, which is the same as is described for grpe-gantt-set-group-state.

grpe-gantt-set-current-time

(*chart*: class grpe-gantt-chart, *current-time*: quantity)

Sets the current time of a gantt chart. The current time is relevant when using grpe-gantt-toggle-group-state to toggle the state on but before the state is toggled off again. If the current time is greater than the time at which the state is toggled on, a bar will be displayed between that time and the current time.

For example, the following two calls produce a bar from 1 to 9 as in the previous examples:

```
call grpe-gantt-toggle-group-state(chart, the symbol person-1,
     the symbol busy, 1);

call grpe-gantt-set-current-time(chart, 9);

// after updating at this point, you'll see a bar from 1 to 9
call grpe-update-chart(chart);

call grpe-gantt-set-current-time(chart, 15);

// after updating at this point, you'll see a bar from 1 to 15
call grpe-update-chart(chart);
```

Class Definitions

This section describes additional attributes on the grpe-chart and grpe-chart-view classes not exposed through the properties dialog.

grpe-chart

Attribute	Description
hotspot-callback-procedure	A procedure that is called whenever the user clicks a point in the chart. The signature of the procedure is: my-hotspot-callback (*chart*: class grpe-chart, *view*: class grpe-chart-view, *subset-index*: integer, *data*: structure, *user-data*: structure, *win*: class g2-window) where: • *subset-index* is the index of the selected subset. • *data* is a structure that contains the X, Y, and Z coordinates of the selected point for a normal chart, or t1, t2, and state for a gantt chart. For an example, see Demo 3. • *user-data* is a structure representing user supplied data which is used primarily for gantt charts. For an example, see Demo 3.
Allowable values:	symbol
Default value:	unspecified
user-callback-procedure	A general callback procedure, which has the same signature as the callback procedure used by g2-ui-create-chart-view. In general, we recommend using the hotspot-callback-procedure instead of the general callback for click events.

Attribute	Description
Allowable values:	symbol
Default value:	unspecified

grpe-chart-view

Attribute	Description
auto-update	When true, automatically updates a native chart whenever the underlying chart object receives new data.
Allowable values:	truth-value
Default value:	true
_initialize-before-update	When true, calls grpe-initialize-view-for-update before any chart update, as described in “Chart Views” on page 51.
Allowable values:	truth-value
Default value:	true
modification-properties	A structure containing additional Gigasoft chart properties that should be applied to the chart view. For details, see Appendix B, “Chart Properties and Enumeration Values” in the *G2 System Procedures Reference Manual*.
Allowable values:	structure
Default value:	structure()

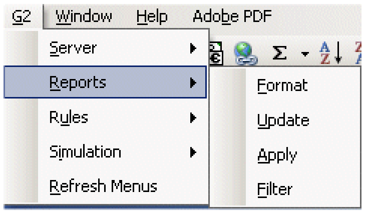
Excel Integration

Describes the Excel integration capabilities of the GRPE module.

Introduction

GRPE provides an Excel add-in called `G2-Excel-VBA.xla` located in the `\g2i\data` directory, which enables G2 to configure Excel workbooks.

To install this add-in in Excel, in Excel, choose Tools > Add-in, click Browse, and choose the G2 add-in. Once the add-in is installed, the following G2 menu and associated toolbar appears in Excel:



Once Excel is connected to G2, G2 can use various APIs to configure Excel workbooks and sheets. The connection between G2 and Excel is based on G2 ActiveXLink.

You can set server name and port number in `G2-Excel-VBA.xla` preliminarily by setting the environment variable G2_SERVER to g2host:g2port (on the computer where Excel is running).

APIs

In the following APIs, *g2com-interface-or-class-name* can be a specific instance to update a single G2 instance, or it can be the `G2Com` interface class name or a subclass to update multiple users with a single call.

In the APIs that specify row and column values as integers, note that the row index and column index both start with 1, not 0; the first row is row 1 and the first column is column 1.

grpe-excel-clear-status-bar
(*g2com-interface-or-class-name*: item-or-value)

Clear the Excel status bar.

grpe-excel-display-message-in-status-bar
(*g2com-interface-or-class-name*: item-or-value, *message*: text)

Displays a message in the Excel status bar.

grpe-excel-post-message
(*g2com-interface-or-class-name*: item-or-value, *message*: text)

Posts a message to the G2 Message Board.

grpe-excel-activate-or-add-sheet
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
activate-sheet: truth-value)

Adds the specified *sheet-name* to the Excel workbook and activates the sheet when *activate-sheet* is true.

grpe-excel-clear-sheet
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
activate-sheet: truth-value)

Clears the specified *sheet-name* from the Excel workbook and activates the sheet when *activate-sheet* is true.

grpe-excel-clear-rows
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
start-row: integer, *end-row*: integer)

Clears the specified rows in *sheet-name* from *start-row* to *end-row*.

grpe-excel-clear-columns
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
start-column: integer, *end-column*: integer)

Clears the specified columns in *sheet-name* from *start-column* to *end-column*.

grpe-excel-delete-sheet
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text)

Deletes the specified *sheet-name*.

grpe-excel-set-sheet-column-data
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
activate-sheet: truth-value, *column*: integer, *start-row*: integer,
end-row: integer, *data*: sequence)

Specifies the column data in the specified *sheet-name* as a sequence, optionally activating the sheet. You provide the *column* whose data to set, and the *start-row* and *end-row*.

grpe-excel-set-sheet-row-data
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
activate-sheet: truth-value, *row*: integer, *start-column*: integer,
end-column: integer, *data*: sequence)

Specifies the row data in the specified *sheet-name* as a sequence, optionally activating the sheet. You provide the *row* whose data to set, and the *start-column* and *end-column*.

grpe-excel-set-sheet-cells-data
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
activate-sheet: truth-value, *cell-references* : sequence, *data*: sequence)

Specifies the cell data in the specified *sheet-name* as a sequence, optionally activating the sheet. You specify *cell-references* as a sequence of text values, which are the cells whose data should be set, and *data* as a sequence of values for each of the cells in *cell-reference*. For example, to set the data of cells A1 and H8 to 2 and 5, respectively, *cell-references* would be sequence("A1", "H8") and *data* would be sequence(2, 5).

grpe-excel-get-sheet-rows-columns-data
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
start-row: integer, *end-row*: integer, *start-column*: integer, *end-column*: integer,
callback-procedure: symbol, *user-data*: text)

Specifies a callback procedure for getting row and column data for the specified *sheet-name* given the *start-row*, *end-row*, *start-column*, and *end-column*. You specify a *callback-procedure,* which executes asynchronously when the data is received. You can specify any required *user-data*, which the *callback-procedure* can access when it executes. The signature for the callback procedure is:

my-callback
(*interface*: class g2com-interface, *user-data*: value, *sheet-name*: text,
number-of-rows: integer, *number-of-columns*: integer, *data*: sequence)

grpe-excel-get-sheet-data
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
start-row: integer, *start-column*: integer, *callback-procedure*: symbol,
user-data: text)

Specifies a callback procedure for getting sheet data for the specified *sheet-name* given the *start-row* and *start-column*. You specify a *callback-procedure,* which executes asynchronously when the data is received. You can specify any required *user-data,* which the *callback-procedure* can access when it executes. See grpe-excel-get-sheet-rows-columns-data for the signature of the callback procedure.

grpe-excel-get-sheet-cells-data
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
cell-references: sequence, *callback-procedure*: symbol, *user-data*: text)

Specifies a callback procedure for getting cell data for the specified *sheet-name* given a sequence of cells specified in *cell-references*. You specify a *callback-procedure,* which executes asynchronously when the data is received. You can specify any required *user-data,* which the *callback-procedure* can access when it executes. The signature for the callback procedure is:

my-callback
(*interface*: class g2com-interface, *user-data*: value, *sheet-name*: text,
number-of-cells: integer, *cells*: sequence, *data*: sequence)

grpe-excel-run-macro
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
activate-sheet: truth-value, *macro-name*: text)

Executes the Excel macro named by *macro-name* on the specified *sheet-name* and optionally activates the sheet.

grpe-excel-setup-g2-menus
(*g2com-interface-or-class-name*: item-or-value)

Displays the G2 menu and toolbar in Excel.

grpe-excel-delete-g2-menus
(*g2com-interface-or-class-name*: item-or-value)

Deletes the G2 menu and toolbar in Excel.

grpe-excel-update-g2-menus
(*g2com-interface-or-class-name*: item-or-value)

Updates the G2 menu and toolbar in Excel.

grpe-excel-set-cell-editable
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text, *row*: integer, *col*: integer, *editable*: truth-value)

Configure the cell in *sheet-name* specified by *row* and *column* to be editable, based on *editable*.

grpe-excel-set-cell-comment
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text, *row*: integer, *col*: integer, *comment*: text)

Creates a *comment* for the cell in *sheet-name* specified by *row* and *column*. Specifying a comment "" clears any comment in the cell.

grpe-excel-set-cell-combo-box
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text, *row*: integer, *col*: integer, *choices*: sequence, *editable*: truth-value)

Creates a combobox in the cell in *sheet-name* specified by *row* and *column* with the specified *choices* and specifies whether the cell is editable, based on *editable*.

grpe-excel-set-cell-color
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text, *row*: integer, *col*: integer, *cell-color*: symbol)

Sets the cell in *sheet-name* specified by *row* and *column* to *cell-color*, which can be any valid G2 color; Excel chooses the closest color in its palette.

grpe-excel-set-cell-text-color
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text, *row*: integer, *col*: integer, *text-color*: symbol)

Sets the color of the text of the cell in *sheet-name* specified by *row* and *column* to *text-color*, which can be any valid G2 color; Excel chooses the closest color in its palette.

grpe-excel-set-cell
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text, *row*: integer, *col*: integer, *cell-struct*: structure)

Sets the contents of the cell in *sheet-name* specified by *row* and *column*, based on *cell-struct*, which has this syntax:

structure
(text-color: *symbol*,
background-color: *symbol*,
bold: *truth-value*,
italics: *truth-value*,
underline: *truth-value*,
comment: *text*,

cell-type: *symbol*
cell-value: *structure*)

where:

- text-color — The text color of the cell.
- background-color — The background color of the cell.
- bold — Whether the text is bold.
- italics — Whether the text is italics.
- underline — Whether the text is underlined.
- comment — A text comment for the cell, where "" clears the cell.
- cell-type — One of these symbols: text-box or combo-box.
- cell-value — A description of the cell value:
  - If cell-type is text-box, a structure with this syntax:

    structure
    (text-value: *text*)
  - If cell-type is a combo-box, a structure with this syntax, which describes the list of choices in the text box and the default selection:

    structure
    (text-sequence: *sequence*,
    selected: *text*)

grpe-excel-set-cell-bold
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
row: integer, *col*: integer, *is-bold*: truth-value)

Sets the cell in *sheet-name* specified by *row* and *column* to be bold or not, based on *is-bold*.

grpe-excel-set-cell-italics
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
row: integer, *col*: integer, *is-italics*: truth-value)

Sets the cell in *sheet-name* specified by *row* and *column* to be italics or not, based on *is-italics*.

grpe-excel-set-cell-underline
(*g2com-interface-or-class-name*: item-or-value, *sheet-name*: text,
row: integer, *col*: integer, *is-underlined*: truth-value)

Sets the cell in *sheet-name* specified by *row* and *column* to be underlined or not, based on *is-underlined*.

Index