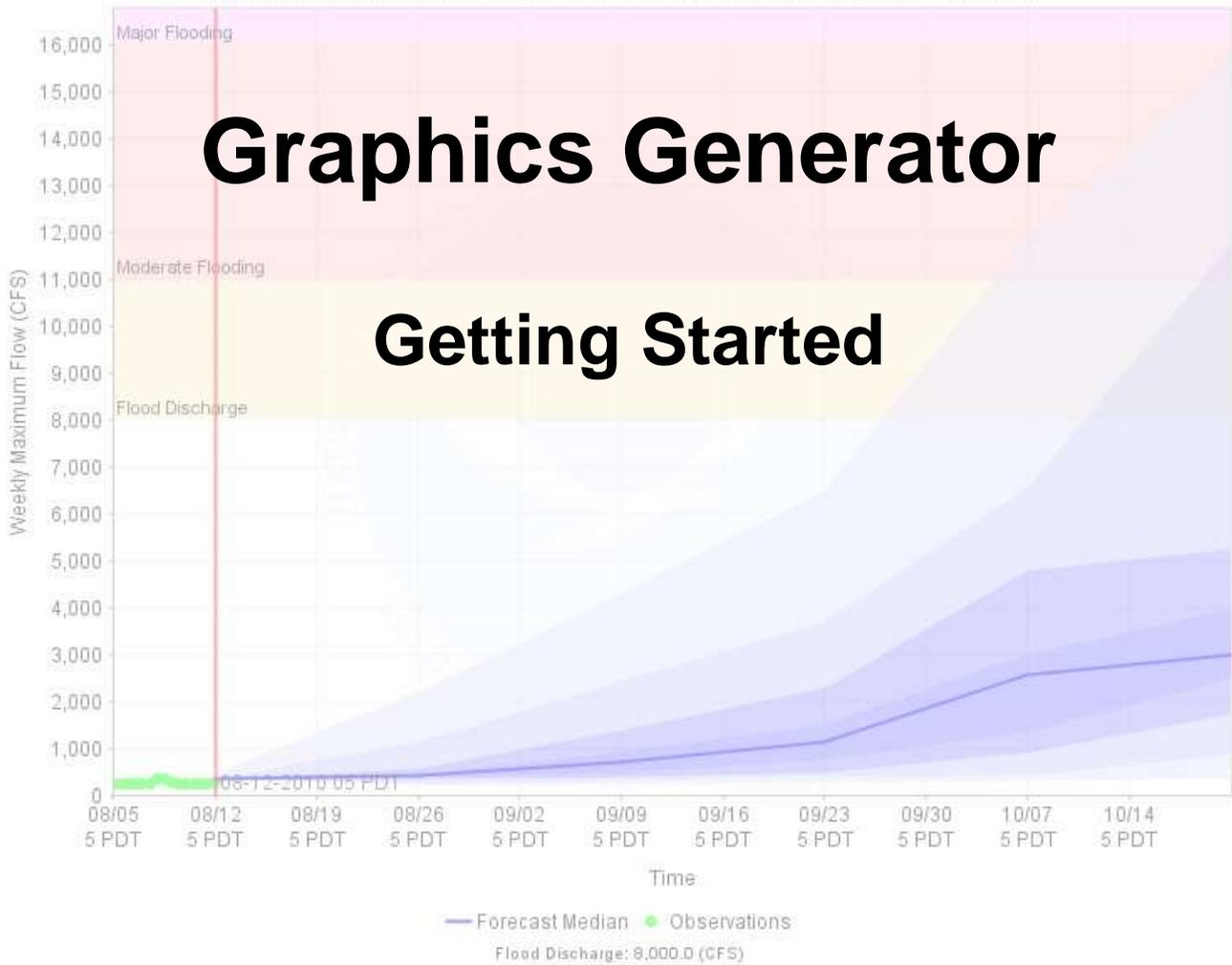


2 Week Peak Flows on the ARGW1 - NF Stillaguamish R Nn Arlington
Forecast for the period 08/12/2010 - 10/27/2010
Cone of Uncertainty Displaying Median, 10%, 25%, 75%, and 90% Exceedance Probabilities



Version: OHD-CORE-CHPS-4.1.a
Release Date: 26 September 2014

Table of Contents

1	Overview	3
1.1	Notation	3
1.2	Terminology	3
2	What is Graphics Generator?.....	4
2.1	What Graphics Generator is Not?	4
3	Why Graphics Generator?	5
4	How Does Graphics Generator Work?.....	7
4.1	Graphics Generator Components.....	7
4.1.1	Graphical User Interface Components.....	7
4.1.2	Component Interaction	8
4.2	Building a Product.....	10
4.2.1	Output Products, Referenced Templates, and Product Templates	10
4.2.2	Product Plug-ins.....	11
4.2.3	Build Process.....	12
4.3	Applying a Product to Many Segments and Times	13
4.3.1	Define the Product for a Product Group.....	13
4.3.2	Arguments	13
4.3.3	Settings	16
4.3.4	Relative Dates	16
4.4	Other Key Concepts	17
4.4.1	Components of a Chart.....	18
4.4.2	Graphics Generator Output	19
4.4.3	Products and Settings Central and Local Areas	20
5	What Next?.....	22

1 Overview

This guide provides an introduction to the Graphics Generator suite of software, describing what it is, why it was written, and how it works. All users should read Sections 2, 3, and 5 before installing the software and any products. Section 4 introduces several key concepts for understanding and using Graphics Generator to construct new products. Though it is also recommended that users read Section 4 before installing the software, as some steps will be more easily understood, it is not mandatory before the products are built or modified.

1.1 Notation

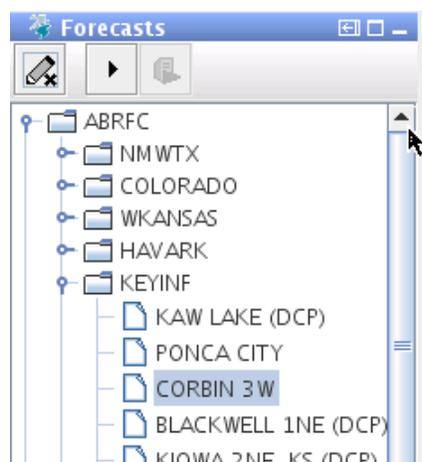
Within this document, the following notation is used:

- All graphical interface components are **Capitalized and in Bold**.
- All XML snippets are in `this font`.
- All command line entries are in `this font`.
- All important terms are *italicized* when first mentioned.

1.2 Terminology

The following terms are used in this document:

- *active forecast segment*: The current active segment, as selected in the **Forecasts Panel** of the CHPS interface and identified by the segment id set in the configuration file `...Config/RegionConfigFiles/Topology.xml`. For example, for ABRFC, the active forecast segment in this case is CBNK1 (Corbin 3W):



2 What is Graphics Generator?

Graphics Generator builds forecast products intended to be used by forecasters during operations, or sent to external customers who need such products for decision making. The components of Graphics Generator allow the user to:

1. Interactively design and build chart-based products with minimal XML configuration.
2. View products for an active forecast segment.
3. Generate chart-based image, XML, or ASCII tabular output product files.

Graphics Generator is designed to replace the product building capabilities of the NWSRFS Ensemble Streamflow Prediction Analysis and Display Program (ESPADP) within CHPS. The software consists of one core component, four graphical user interface components, and a model adapter:

- Graphics Generator Engine: The core component of Graphics Generator, it is responsible for building products.
- **GraphGen Tree Panel:** A graphics user interface component used to manage the products and settings, including deleting, renaming, importing and exporting.
- **GraphGen Editor Panel:** A graphics user interface component used to build and modify products.
- **GraphGen Thumbnails Panel:** A graphics user interface component used to view a summary of products for an active forecast segment and manage the segment's settings.
- **GraphGen Viewer Panel:** A graphics user interface component used to view one product.
- Graphics Generator Model Adapter: A model adapter that allows for generating product files via a CHPS workflow.

More information on the interaction of these components and how a product is constructed is provided in Section 4.

2.1 *What Graphics Generator is Not?*

Graphics Generator is not designed for operational analysis of ensemble forecasts. Though the **GraphGen Editor Panel** can be used for analysis, it is designed to facilitate building products. As such, analyzing a forecast ensemble via the **GraphGen Editor Panel** will require more interaction (i.e., more clicks) than should be necessary.

Graphics Generator is not a replacement for CHPS time series viewing and editing components. Though it shares much of the same functionality with some CHPS components, it is not as fast (due to the PI-service being used), does not allow for editing data, does not allow for applying transformations on the fly, and no time series computed via Graphics Generator are stored in the CHPS local data store.

3 Why Graphics Generator?

CHPS is built on top of the FEWS software package, and that package is not able to perform many of the calculations or generate some of the plots required by ESPADP. For example, it cannot be used to compute the number-of-days-to aggregations available in ESPADP or generate probability plots.

In some cases where computations can be performed, the amount of XML configuration required is significant, and possibly even prohibitive. For example, to compute a 24-hour minimum or maximum time series from a 6-hour time series the 6-hour time series must be shifted by 6, 12, and 18 hours, to yield four time series (counting the original time series). Each of those time series must then be sampled to yield time series that contain only one value per day, and those time series are then put into an equation to compute the minimum or maximum value. Such configuration would only work for converting from 6-hours to 24-hours and for a single data type. In Graphics Generator, such an aggregation requires only a few clicks (create an aggregation, specify its type as minimum or maximum, and specify its step as 24-hours) and is readily applicable across all data types.

The following table provides a listing of capabilities ESPADP provides, and identifies if and how CHPS only (without Graphics Generator) and CHPS with Graphics Generator installed can replicate those capabilities:

ESPADP Capability	CHPS (Only)	CHPS w/Graphics Generator
Aggregations	Configuration (Transformations)	Interactive
Instantaneous/Sum/Mean	Yes (single transformation)	Yes
Min/Max	Yes (multiple transformations)	Yes
Accumulative	Yes (multiple transformations)	Yes
Counts (e.g., NDTO)	No	Yes
Computations	Configuration (Transformation)	Interactive
Quantiles	Yes (?)	Yes
Probabilities	Yes	Yes
Moments	Yes	Yes
Probability Plot	No	Yes
Distribution Fitting	No	Yes
Year-Weighting	No	Yes
Plot Types	Configuration	Interactive
Spaghetti	Yes	Yes
Histogram	Yes	Yes

ESPADP Capability	CHPS (Only)	CHPS w/Graphics Generator
Box-and-Whisker	Yes	Yes
Scatter-and-Line	Yes	Yes
Appearance Modification	Configuration	Interactive
Thresholds	Yes	Yes
Rating Curve Axis	Yes	Yes
Label/Text Font	Yes	Yes
Output Products	Configuration	Interactive
Images	Yes	Yes
Tables	Yes	Yes
Time Series (Datacard)	Yes (PI-timeseries XML)	Yes (PI-timeseries XML)

The following table compares general appearance modification capabilities available in CHPS (only) and CHPS with Graphics Generator installed:

Capability	CHPS (Only)	CHPS w/Graphics Generator
Area Between Lines Plot Type (useful for cone of uncertainty)	Partial (can display the area between a min and max)	Yes
Line Controls	Yes	Yes
Line style (solid, dashed, etc.)	Yes (via line style choices)	No (small enhancement required)
Line thickness	Partial (via line style choices)	Yes (full control)
Scatter Point Controls	Yes	Yes
Scatter fill color	No	Yes
Scatter text	Yes	No
Subplot Weights	Yes	Yes
Inverted Axis (range/vertical axis displayed upside down)	Yes	Yes
Ax7is Limits/Scale	Partial (scale unit control; control over upper/lower margins)	Yes (full control)
Background Image	No (for reports, yes)	Yes
Background Color	Yes	Yes
Legend Position	Partial (right, bottom only)	Yes (right, bottom, left, top)
Subtitles (text displayed next to the chart on any side)	No	Yes

4 How Does Graphics Generator Work?

Graphics Generator works within the CHPS and FEWS framework in order to build products. The Graphics Generator Engine acquires time series from the FEWS PI-service as input; the four graphical user interface components are FEWS explorer plug-ins, seamlessly integrating within the CHPS interface; and the model adapter is run as a standard general adapter via a CHPS workflow. Details about how Graphics Generator works are provided below.

4.1 Graphics Generator Components

As described above, there are six components to Graphics Generator: an engine, four graphical user interface components, and a model adapter. The graphical user interface layout and how the components interact are described below.

4.1.1 Graphical User Interface Components

The Graphics Generator graphics user interface components are FEWS explorer plug-ins displayed within the CHPS user interface. The components are as follows (a screenshot is shown below):

1. **GraphGen Tree Panel**

Typically displayed on the left along with the CHPS **Data Viewer** and **Forecasts Panels**, this panel allows the user to manage and select Graphics Generator products and templates for editing or viewing, as well as copying and deleting products and moving products between segments and groups. It also provides access to tools for importing and exporting products and modifying settings.

2. **GraphGen Thumbnail Panel**

Typically displayed on the right along with the CHPS **Plot Overview** and **Forecaster Help Panels**, this panel allows for the user to view thumbnails of products for the current active forecast segment and select products for viewing. It possesses the same look-and-feel as the CHPS **Plot Overview**.

3. **GraphGen Editor Panel**

Typically displayed in the center along with the **Map, Plots, Topology**, and other panels, this panel allows for the user to create or modify Graphics Generator products.

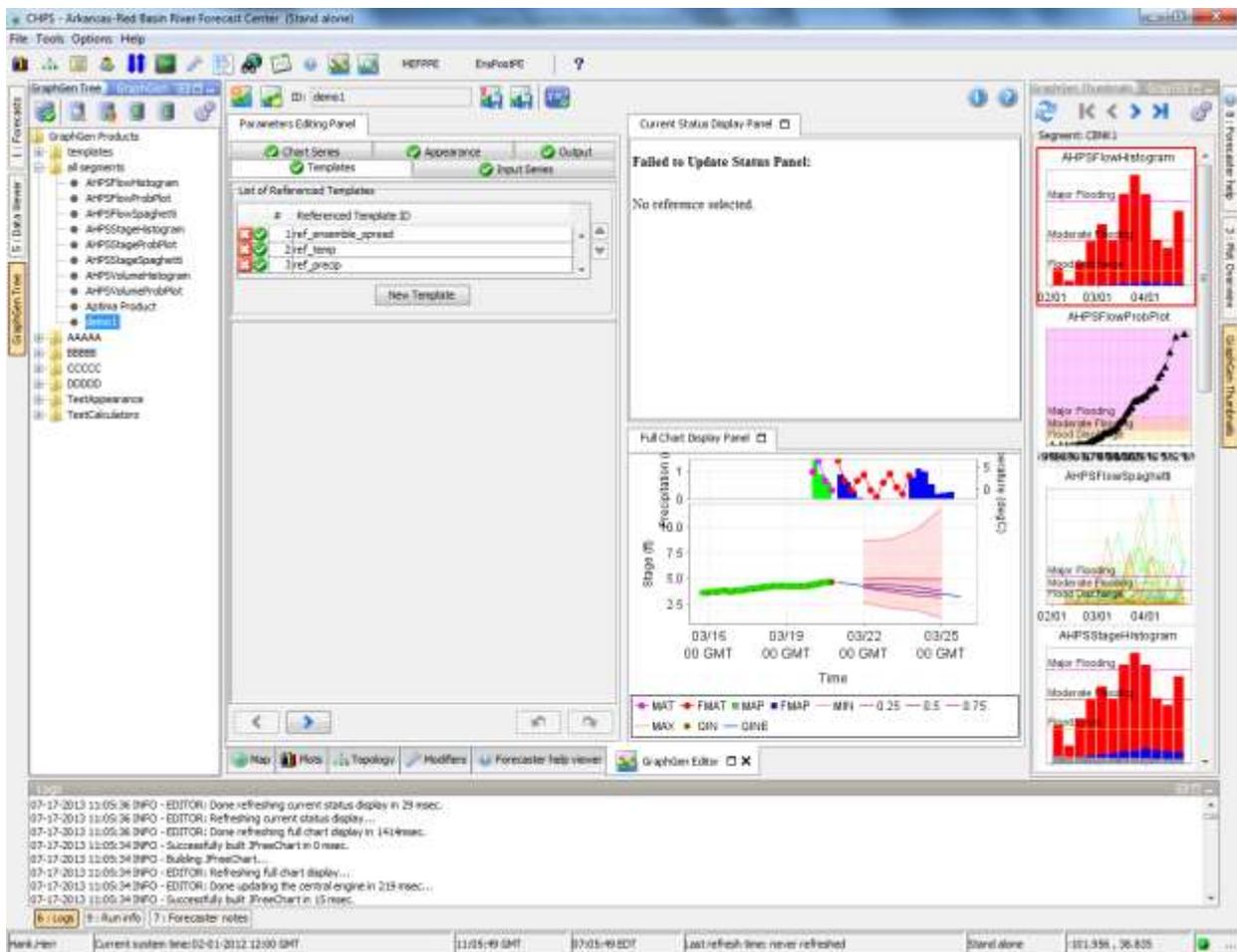
4. **GraphGen Viewer Panel**

Typically displayed in the center, this panel allows for the user to view Graphics Generator chart products.

As with any CHPS interface component, a tab or toolbar button must be clicked to open a component. Furthermore, the components can be detached or reattached to the main interface, or be minimized.



All log messages generated by the four graphical user interface components are displayed in the CHPS **Logs Panel** and log.txt file, as is standard for all CHPS components.



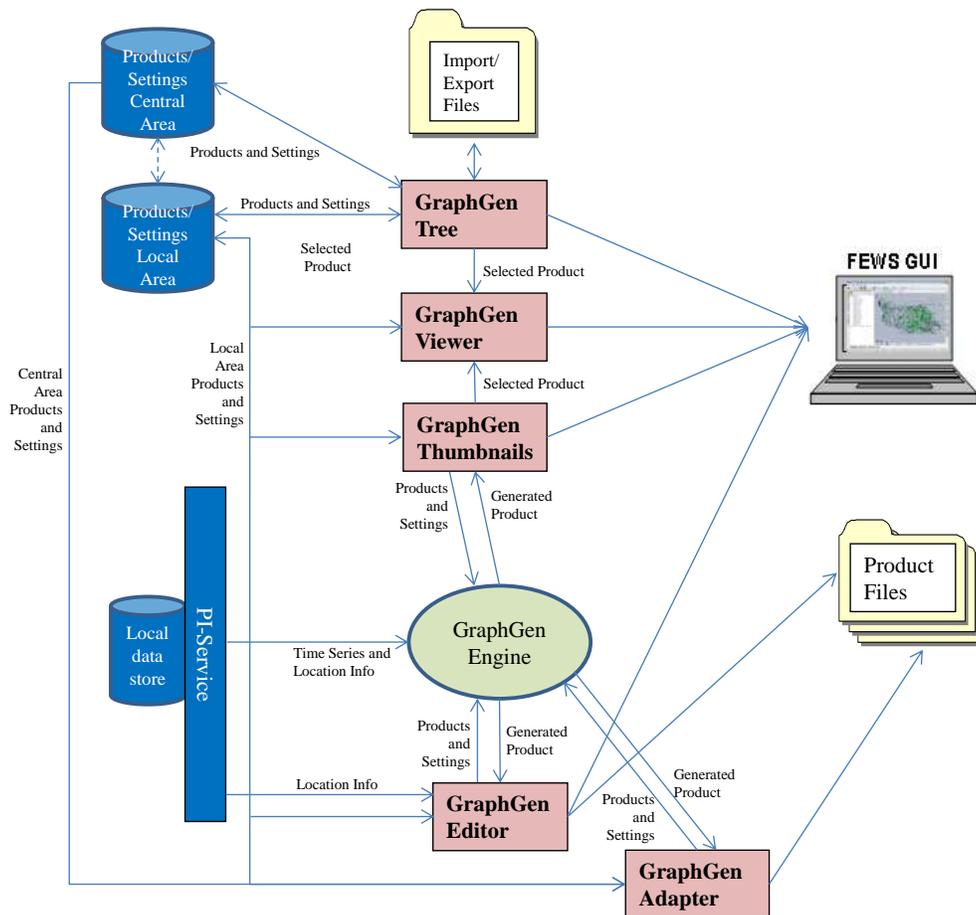
4.1.2 Component Interaction

The interaction of the Graphics Generator components with the FEWS PI-service and central and local area product and settings storage is diagrammed below, and is summarized as follows:

- A products and settings local or central area is accessed by all Graphics Generator components to acquire product definitions and settings. A local area is specific to a CHPS stand-alone and stores changes to products and settings made within that stand-alone. Changes are promoted to the central area via a button click in the **GraphGen Tree Panel**. Products and settings in the central area are not modified directly by any CHPS stand-alone; rather, it stores promoted products and settings for use in non-interactive product generation via the GraphGen Model Adapter, which executes the Graphics Generator Engine.
- The Graphics Generator Engine acquires product definitions and settings from either the central or local area, acquires time series and location information from the FEWS PI-service, and generates products files (images, XML, and ASCII table files).
- The **GraphGen Tree Panel** acquires product definitions and settings from the local area and displays them as a tree to the user. It also provides tools to promote products and settings to the central area.

- The **GraphGen Editor Panel** acquires the selected product from the **GraphGen Tree Panel**, acquires settings from the local area, acquires time series and location information from the FEWS PI-service, calls the Graphics Generator Engine to build the edited product, and displays that product to the user.
- The **GraphGen Thumbnails Panel** acquires product definitions and settings from the local area and calls the Graphics Generator Engine to build products, which it then turns into thumbnails and displays to the user.
- The **GraphGen Viewer Panel** acquires the selected product from the **GraphGen Thumbnails Panel** or **GraphGen Tree Panel**, acquires the settings from the local area, and calls the Graphics Generator Engine to build the selected product. If the product was selected via the **GraphGen Thumbnails Panel**, the panel is accessed for the already prepared chart so that the viewer does not rebuild it from scratch.
- The GraphGen Model Adapter acquires product definitions and settings from either the central or local area (depending on run file properties), calls the Graphics Generator Engine to build the product, and generates output files.

All FEWS PI-service interactions are handled by wrappers for efficient access. For example, a wrapper ensures that input series are only loaded once when generating a product even if multiple templates refer to them. The diagram is as follows:



4.2 Building a Product

When designing and building Graphics Generator products, it may be useful to understand how the products are constructed. This section first presents the concepts of templates and products, as well as product plug-ins, which are integral to understanding how products are built, and then describes the process of building a product.

4.2.1 Output Products, Referenced Templates, and Product Templates

Core to the understanding of how Graphics Generator builds a product are the concepts of output products, templates, referenced templates, and product templates.

- *output product*: the ultimate output of Graphics Generator: it is a chart image or ASCII output file, such as a PI-timeseries XML file or tabular ASCII file.
- *template*: a list of instructions for Graphics Generator on how to construct a chart, specifying input series to use, calculations to perform, and the appearance.
- *referenced template*: a template that includes no additional information about how to generate output products. It is meant to be referred to from within instructions provided by a *product template*.
- *product template*: a template combined within information output products to create based on the generated chart. The template instructions in a product template can refer to referenced templates in order to construct the chart.

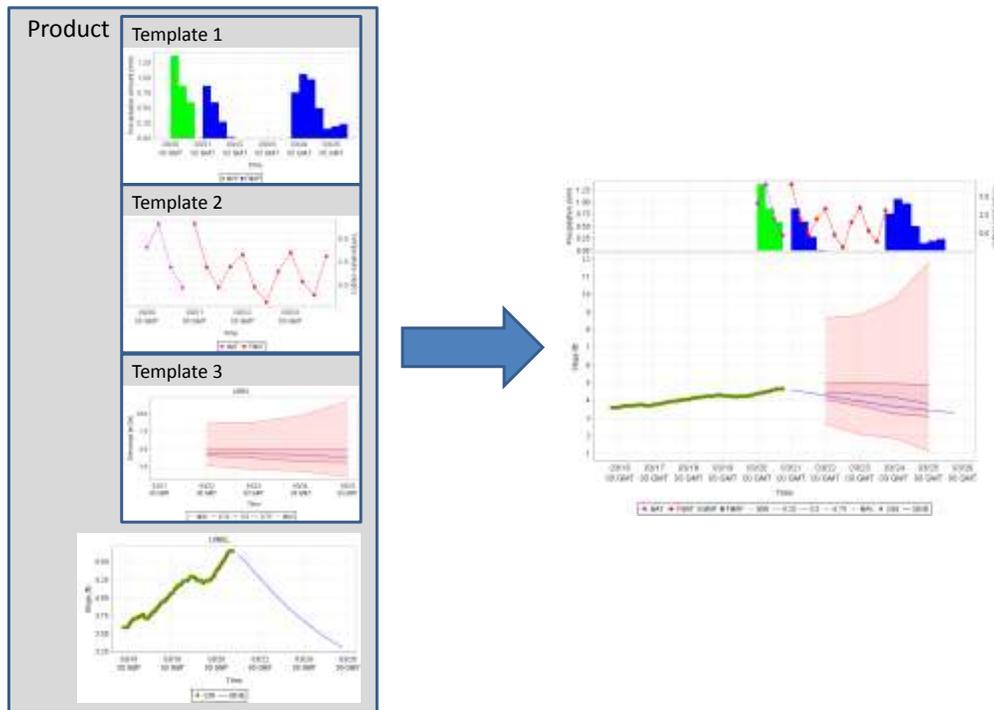
IMPORTANT!

Throughout this rest of this guide, all documentation involving Graphics Generator, and the Graphics Generator interface components, the following short-hand is used:

- *referenced template* = “template”. When seeing the word “template” by itself, it is assumed to refer to a *referenced template*.
- *product template* = “product”. When seeing the word “product” by itself, it is assumed to refer to a *product template*.

A referenced template is identified by a template id; a product template is referred to by both the template id and a CHPS segment id. Alternative, the CHPS segment id can refer to a *product group*, which allows for grouping product templates so that an output product may be generated for any forecast segment for which the specified group is included. Note that the product group “all segments” is automatically included for every segment for which products are constructed.

A product (product template) can include chart series computed via templates (referenced templates), as well as those computed directly in the product. For example, consider the following product:



In this case, the product, shown on the right, includes three templates (labeled Template 1, 2, and 3) and two chart series computed directly in the product (the bottom chart on the left).



Though it is not recommended because of the complexity introduced by doing so, it is possible for a template to refer to other templates.

4.2.2 Product Plug-ins

The Graphics Generator makes use of five types of *plug-ins* to generate a graphic:

1. *Input Series Provider Plug-in*: provides time series that can be used to build chart series displayed in a Graphics Generator chart.
2. *Aggregator Plug-in*: aggregates a time series.
3. *Calculator Plug-in*: accepts time series as input and generates output series for display in a chart.
4. *Appearance Modifier Plug-in*: modifies the appearance of a Graphics Generator chart.
5. *Output Generator Plug-in*: accepts a chart and its underlying data and parameters as input and generates output of some kind; e.g. images, tabular text files, and XML files.

Usage of a plug-in architecture allows for easy extension of the capabilities of Graphics Generator, potentially even by external users. The plug-ins are accessed throughout the build process. Parameters of the plug-ins are specified in the product definition and can be set via

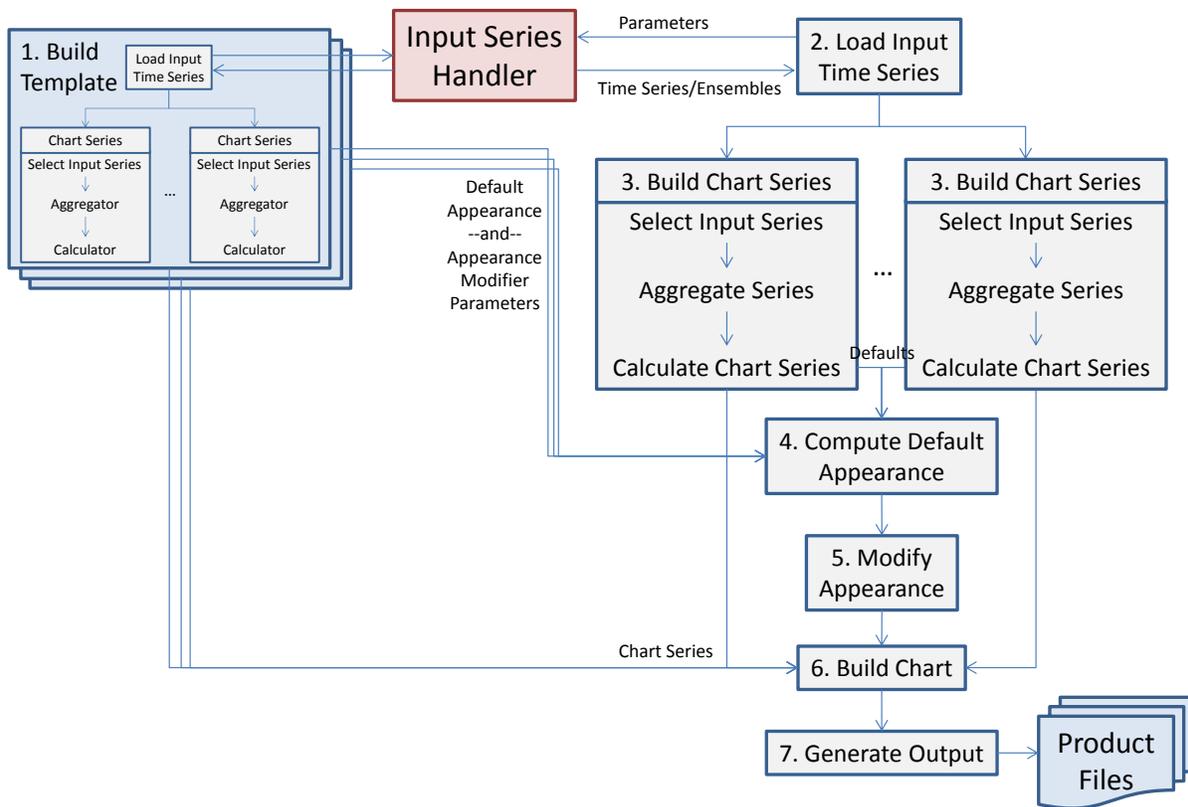
components of the **GraphGen Editor Panel**; see the *Graphics Generator Reference Manual* for more information.

4.2.3 Build Process

The process of building a chart, as executed in the Graphics Generator Engine, is as follows:

1. Build the chart series for each template, in order.
2. Build a default appearance based on the templates, with the parameters for a template overriding previously included templates. For example, if each template specifies a plot title, then it is the last template's plot title that defines the default appearance.
3. Build the product-specific chart series as follows:
 - a. Load all specified input series.
 - b. For each defined chart series, do the following:
 - i. Select time series from the loaded chart series.
 - ii. Aggregate the selected time series.
 - iii. Compute the chart series and default chart series appearance.
4. Compute the overall default appearance combining the output from Step 2 with the appearance of chart series built in Step 3.
5. Modify the appearance based on product parameters.
6. Build the final chart.
7. Generate output based on the chart (image file, XML, or ASCII tabular output).

This process is diagrammed below:



4.3 Applying a Product to Many Segments and Times

An important aspect of any designed Graphics Generator product is that it be generalizable; i.e., it can be applied for many, or all, forecast segments and for many system times. To facilitate this, product instructions can make use of arguments, settings, and relative dates.

4.3.1 Define the Product for a Product Group

As stated in Section 4.2.1, products are defined for individual segments or for groups. For a product to be applicable to many segments, it must be defined for a *product group* and that group must be included as being *included* for each segment for which the product is to be constructed. The group “all segments” is automatically included for all segments for which products are constructed. Product groups may be included for a segment via the **Modify Settings Dialog**, which is accessible via the **GraphGen Tree Panel** and **GraphGen Thumbnails Panel**; see the *Graphics Generator Reference Manual*.

4.3.2 Arguments

Arguments are what allow a product defined for a group to be applied to multiple segments, presumably with different locationIds, location names, thresholds, and possibly even data types.

An *argument* is a mapping of a name to a value and has many uses. For example, arguments can be used to specify file names of output to generate, location ids of time series to load, and labels

within the plot. By referring to arguments, a template or product can be designed such that, given a different set of arguments, it can be applied to a different location or forecast point.

Within labels, file names, and so on, an argument is indicated by the “@” symbol before and after the name of the argument. For example, a template can be designed to display the time series loaded for location id “@defaultLocationId@”. Then, to apply the template to location AAAAA, specify an argument to the template with name “defaultLocationId” and value “AAAAA”. To apply the same template to location BBBBB, do the same but set the argument value to “BBBBB”.

4.3.2.1 *Product Specific and Predefined Arguments*

Arguments may be either predefined or product specific. A predefined argument is one that is made available to any product created for a specific segment. The predefined arguments for a segment are configurable within the **Modify Settings Dialog**, which is accessible via the **GraphGen Tree Panel** and **GraphGen Thumbnails Panel**; see the *Graphics Generator Reference Manual*. There are six predefined arguments automatically included for any segment:

- `overrideActiveSegmentId`: Overrides the active forecast segment id in the CHPS interface, or “-undefined-” if none. This is the segment id used by the Graphics Generator to load the other predefined arguments configured for that segment.
- `defaultLocationId`: By default it has the same value as `overrideActiveSegmentId`.
- `defaultParameterId`: “QINE” by default.
- `defaultEnsembleId`: “ESP” by default.
- `handbook5Id`: The five-letter Handbook-5 id for the active forecast segment. By default, it is the first five letters of the active forecast segment id.
- `systemTime`: The system time, T0, to use where appropriate; by default, it is the CHPS active system time.

Additional predefined arguments are defined based on included product groups; all predefined arguments defined for any included groups are available as a predefined argument for the segment.

Product specific arguments are defined by the user and tailored for the specific product. Whereas any predefined argument has a default initial value, a product specific argument does not. This means all product specific arguments must be provided before a template can be included in a product.

4.3.2.2 *Argument Functions*

An *argument function* is an argument that requires parameters. The parameters are specified within parentheses and are semicolon delimited, immediately after the argument function name and before the closing ‘@’. For example:

“@relativeDateToText(T0; MM/dd/yyyy HH:mm:ss z; GMT)@”.

Within the Graphics Generator, the following are argument functions:

- **latitude**: Returns the latitude of a location as defined in the Locations.xml CHPS configuration file. The parameter is the location id, and it may include arguments. For example, this is valid: “@latitude(@defaultLocationId@)@”.
- **locationDescription**: Returns text specifying the description for the location as defined in Locations.xml. The parameter is the location id, and it may include an argument.
- **locationShortName**: Returns text specifying the short name for the location as defined in Locations.xml. The parameter is the location id, and it may include an argument.
- **longitude**: Returns the longitude of a location as defined in Locations.xml. The parameter is the location id, and it may include an argument.
- **relativeDateToText**: Returns text corresponding to a relative date in a specified format.

The parameters are:

1. the relative date string (e.g., “T0 + 90 days”)
2. the date format (e.g. “MM/dd/yyyy HH:mm:ss z”)
3. the time zone

See Section 4.3.4 for information on the date format and valid time zones.

- **thresholdValue**: Returns text specifying a threshold value in the units of the input series provider that provided the threshold. The parameters are:
 1. the identifier of the threshold, viewed in the **Parameters Editing Panel** of the **Appearance Panel** when “Thresholds” is selected from the **Plug-in Tree** (see Section 12 of the *Graphics Generator Reference Manual*).
 2. the number of decimal places to display in the returned text

Argument functions work exactly like arguments in every other way. For example, the **relativeDateToText** argument above would return text corresponding to the system time, T0, in the format specified by the date format string, “MM/dd/yyyy HH:mm:ss z”.



Do not use semicolons in the parameter values of an argument function! They must only be used to separate the parameter values in the list.

4.3.2.3 Global and Template Arguments

There are two levels of arguments: global and template. *Global arguments* are associated with a product and can be accessed by referenced templates or by the product specific parameters. *Template arguments* are associated with a specific referenced template, overriding any global arguments with the same argument names and being used in conjunction with the template to build the template chart series.

4.3.2.4 Arguments within Arguments

An argument can be referenced within the value of another argument or parameters of an argument function. An argument function can be referenced in the value of an argument, but

cannot be referenced in the parameters of another argument function. Reference cycles are not allowed; i.e., if argument a is defined using argument b, then argument b cannot be defined using argument a.

4.3.3 Settings

Settings define default behavior within the Graphics Generator. Specifically, the settings include:

1. A listing of product groups to include for a segment.
2. A definition of predefined arguments associated with a segment.
3. Which of the products are to be displayed as thumbnails within the **GraphGen Thumbnails Panel** for a segment (by default, all are displayed).

These three types of settings can be defined for any segment, while the latter two can also be defined for “all segments” (the product group “all segments” may not, itself, include other product groups). For any segment, the predefined arguments are a combination of those defined for “all segments”, overridden by those defined for included product groups (in order of inclusion), and further overridden by those defined specifically for the segment. The product visibility is defined in the same manner: the default visibility is defined for “all segments”, overridden based on included product groups, and then overridden by segment specific settings.

All settings are modified via the **Modify Settings Dialog**, which is accessible via the **GraphGen Tree Panel** and **GraphGen Thumbnails Panel**; see the *Graphics Generator Reference Manual*.



Settings are stored in the local data store in the same manner as products.

4.3.4 Relative Dates

Dates within the Graphics Generator can be fixed or relative. Fixed dates are constant dates that do not change regardless of the forecast scenario (e.g. “02-02-2010 00:00:00 GMT”). They are of limited usefulness in creating reusable charts or templates, because they do not change when the system time moves forward. Relative dates, however, can be defined relative to the system time or to a time series start time and end time, making them more generally applicable. A relative date follows this format:

```
<basis date> <+/-> [<quantity> <unit>] [<quantity> <unit>] ...
```

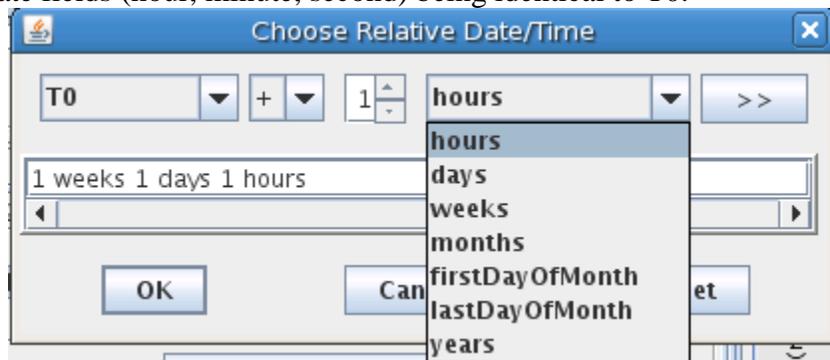
In general, the basis date can be one of the following:

- “T0”: The CHPS system time, defined in the lower left corner of the CHPS interface:

Current system time: 10-15-2009 12:00 GMT

- “tsT0”: Given a base time series, the system time used when the time series was generated. Useful for defining the aggregation period of past forecast time series.
- “tsStartTime”: Given a base time series, the start time of the time series.
- “tsEndTime”: Given a base time series, the end time of the time series.

The units may be either “years”, “months”, “weeks”, “days”, or “hours”. Two special units that allow for defining irregular time steps are also available: “firstDayOfMonth”, “lastDayOfMonth”. For example, “T0 + 1 firstDayOfMonth” will return the first first-day-of-a-month following T0 with all other date fields (hour, minute, second) being identical to T0. Similarly, “T0 + 2 firstDayOfMonth” will return the second last-day-of-a-month following T0 with all other date fields (hour, minute, second) being identical to T0.



4.3.4.1 Examples

The following are examples of relative dates and corresponding evaluated dates. For these examples, the system time is 02/01/2010 00:00:00 GMT.

- | | |
|---|-------------------------|
| • “T0 + 1 day” | 02-02-2010 00:00:00 GMT |
| • “T0 + 1 week 6 hours” | 02-08-2010 06:00:00 GMT |
| • “T0 + 1 day 3 days 12 hours” | 02-05-2010 12:00:00 GMT |
| • “T0 + 1 firstDayOfMonth” | 02-01-2010 00:00:00 GMT |
| • “T0 + 1 lastDayOfMonth” | 03-31-2010 00:00:00 GMT |
| • “T0 + 2 firstDayOfMonth 2 lastDayOfMonth” | 04-30-2010 00:00:00 GMT |

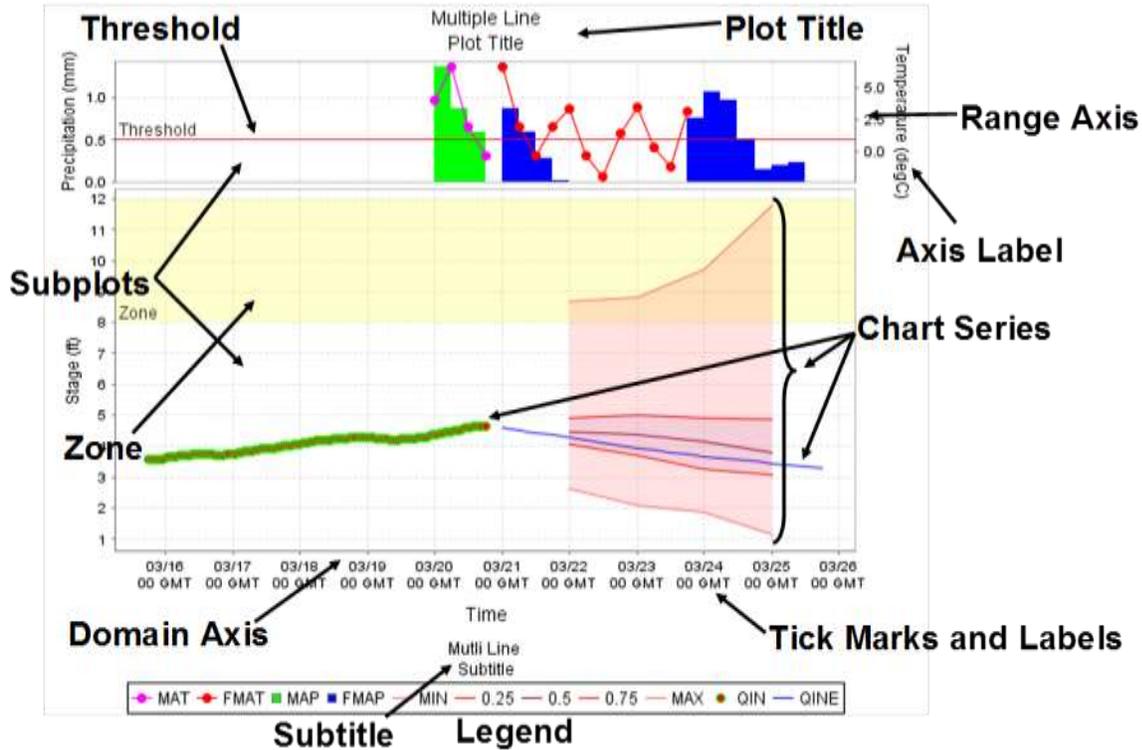


When processing a relative date, the firstDayOfMonth unit is always processed before the lastDayOfMonth unit.

4.4 Other Key Concepts

4.4.1 Components of a Chart

The main functions of the Graphics Generator are to build and display X-Y charts, and generate output files that are derived from the chart. Thus, the user should understand the components of a chart before using Graphics Generator, as diagrammed here:



4.4.1.1 Chart Series

A *chart series*, as the term is used herein, is a set of displayable X-Y *series* computed by a single calculator plug-in and displayed within a chart. It is calculated based on four items:

1. *Selected time series*: Time series selected as input to the chart series computation process.
2. *Aggregators*: A sequence of aggregations to be performed by aggregator plug-ins, with the output of the previous plug-in becoming the input to the next plug-in.
3. *Calculator*: A calculation to be performed by a calculator plug-in that yields a set of series to be plotted on the chart.
4. *Series drawing parameters*: Parameters defining instructions for how to display the chart series in the chart (plot types, colors, shapes, etc).

4.4.1.2 Subplot

A *subplot* is any plot within the charting area. All subplots share the same x axis, but can have different y axis. A chart consists of any number of subplots, with the first subplot being the top subplot.

4.4.1.3 *Domain Axis*

The *domain axis*, or x-axis, is the horizontal axis along the bottom of the chart. Typically, the axis will be time-based, but there are some cases where it uses other units. The axis will include tick marks at regular intervals, tick mark labels, and a one-line axis label.

4.4.1.4 *Range Axis*

A *range axis*, or y-axis, is a vertical axis on either side of a subplot. Typically, the axis will be numerical. The axis will include a label, tick marks at regular intervals, and tick mark labels.

4.4.1.5 *Titles*

The *plot title* is a multi-line title at the top of the plot. A *subtitle* is an extra annotation on the plot that can be positioned above, below, or on either side of the plot. If above the plot, it will be positioned between the plot title and the top subplot.

4.4.1.6 *Legend*

The chart *legend* displays the series plotted within the chart. For each series, it will include a name and an image that will allow the viewer to recognize the series within the chart (usually a line and/or symbol of the same color as what is in the chart).

4.4.1.7 *Threshold*

A *threshold* is a labeled line within a subplot marking a significant value, such as a time or flood stage.

4.4.1.8 *Zone*

A *zone* labeled area within a subplot marking some significant range of values, such as the area above flood stage.

4.4.2 **Graphics Generator Output**

The Graphics Generator is capable of generating any output for which an output generator plug-in is available. However, all of the output must be derived from the chart series contained in a chart constructed by the Graphics Generator charting engine. For example, to output time series in a tabular format, the user must define a chart that displays the desired chart series and specify the appropriate output generator plug-in. To output the minimum, 10%, median, 90%, and maximum values drawn from an ensemble forecast in tabular format, the user must define a chart

in Graphics Generator in which those five output values are computed and displayed and specify the appropriate output generator plug-in.

4.4.3 Products and Settings Central and Local Areas

The Graphics Generator stores products and settings in XML files placed on the file system in two areas:

1. The *local area* is used to store products and settings for access by Graphics Generator components installed within an associated stand-alone. Changes made to products or settings within that stand-alone are recorded in the local area. The location of the local area is always as follows (using CHPS global property notation): %REGION_HOME%/Models/graphgen.
2. The *central area* is used to store products and settings that are considered ready for production. The products and settings in the central area should never be edited directly. The location of the central area is defined by the CHPS global property ohdGraphgenCentralDir.

In both areas, the products and settings are stored in an XML with the name OHD_GRAPHGEN_PRODUCTS_AND_SETTINGS.xml. The **Upload Products** and **Download Products** buttons in the **GraphGen Tree Panel** can be clicked to copy the products and settings from the local area to the central area and vice versa, respectively:



Whenever the **Upload Products Button** is clicked, after copying the file from the local area to the central area, a date-stamped copy of the file will be created:

OHD_GRAPHGEN_PRODUCTS_AND_SETTINGS.xml.yyyyMMdd_HHmms

Up to 50 such files may be recorded in the central area directory. These files provide a crude method for recovering modified or removed referenced and product templates (see the *Graphics Generator Tips and Troubleshooting Guide*). This simple model does not provide robust synchronization capabilities between products defined in multiple stand-alones. Hence, it is recommended that users follow these guidelines:

- Only use one stand-alone at any time to edit Graphics Generator products and settings.
- Upon first opening the editing stand-alone, be sure that the local area products and settings are updated to match the central area; *download* the products and settings by clicking on the **Download Button** in the **GraphGen Tree Panel**. This download will happen automatically when a stand-alone with Graphics Generator properly installed is executed for the first time.
- Do not download the products and settings from the central area again unless you wish to discard working changes. Downloading will overwrite any changes made in the local area.

- When done editing products and settings, *upload* them to the central area by clicking the **Upload Button** in the **GraphGen Tree Panel**. After this point, downloading products and settings from the central area can be done as needed.
- When run as part of a scheduled workflow, the GraphGen Model Adapter should always be configured to run using products and settings defined in the central area.
- When testing the execution of the GraphGen Model Adapter for products being edited, it must be configured to run using the editing stand-alone's local area or it will not have access to locally edited products.

If these guidelines are not followed, conflicts may occur when editing products or running the adapter.

5 What Next?

After reading this manual, the next step is to install and use Graphics Generator. Provided with this release is the following documentation:

- *Graphics Generator Installation Guide*: Step-by-step instructions for installing the basic Graphics Generator package, including the four graphical interface components.
- *AHPS Products Installation Guide*: Step-by-step instructions for installing the OHD-delivered AHPS products. Instructions are provided for XML configuration, importing the products and settings, and tailoring those products and settings to the needs of the RFC and its segments.
- *Graphics Generator Reference Manual*: Detailed guide to the many parts of the Graphics Generator components, including all four graphical interface components and the model adapter.
- *Graphics Generator Tips and Troubleshooting*:

It is recommended that you do the following:

1. Performing the installation outlined in the *Graphics Generator Installation Guide* and verify that it works.
2. Perform the steps provided in the *AHPS Products Installation Guide* to set up some default, starting products. Verify that it works.
3. Read the Overview of the *Graphics Generator Tips and Troubleshooting*, which describes which sections therein should be read before creating products.
4. Play around with the four interface components of Graphics Generator as desired; the best way to learn the software is to try to create your own products and gain hands-on experience.