Automatic Program Generation
(APG™) Option
# Table of Contents

1.0 Introduction ...........................................................................................................4  
2.0 Installation .............................................................................................................5  
3.0 Quick Start .............................................................................................................6  
4.0 Creating APG Modules .......................................................................................10  
  4.1 APG Dynamic Link Library (DLL) ..................................................................10  
  4.2 APG Executable (EXE) ....................................................................................10  
  4.3 APG Build ........................................................................................................ 11  
    4.3.1 Target Processors ................................................................................11  
    4.3.2 Comments .............................................................................................. 11  
    4.3.3 Output File ............................................................................................ 11  
    4.3.4 APG Build .............................................................................................. 12  
    4.3.5 Auto Replace MetaSystem ...................................................................12  
    4.3.6 Finish ..................................................................................................... 12  
5.0 Working With an APG Executable .....................................................................13  
  5.1 APG EXE Controls........................................................................................... 13  
  5.2 APG EXE Execution ........................................................................................ 15  
  5.3 APG Results ..................................................................................................... 16  
    5.3.1 Importing APG Sink Data in SystemView .........................................16  
    5.3.2 APG Sink File Management................................................................17  
  5.4 User Code Support ........................................................................................... 17  
6.0 Troubleshooting ..................................................................................................18  
Appendix A. Microsoft Visual C++ Installation ......................................................19  
Appendix B. Custom Compilation ...........................................................................20  
Appendix C. APG Components ...............................................................................21
1.0 Introduction

The SystemView Automatic Program Generation (APG) option is a tool for creating 32-bit Windows executable programs from SystemView. The option allows the creating of programs from complete system designs. It also allows the creating of user code, and compatible Dynamic Link Libraries (DLLs), from SystemView MetaSystems. In this manual the term APG refers to EXEs or to DLLs created from SystemView using the APG option.

APG modules are completely self-contained, and therefore are easily transferable between computers. They exactly represent the corresponding SystemView simulations while running much faster, which is the primary purpose, to greatly accelerate simulation times. APG results can be examined in the SystemView Analysis Window, or saved as various type files to provide easy importing into other analysis packages.

The SystemView Automatic Program Generation (APG) option works only with Microsoft Visual C++ tools. The APG option works by generating a control program in C, that corresponds to the execution sequence of the SystemView model, then invoking the C++ compiler to compile it, and then invoking a linker to include the token objects from a pre-compiled library.
2.0 Installation

The SystemView Automatic program Generation (APG) option is installed automatically with SystemView. If the APG option license was purchased, access to the software is gained immediately after installing SystemView. If the APG option license has not been purchased, it may be enabled over the telephone, by contacting Eagleware-Elanix or an authorized SystemView distributor.

Two major components are necessary to use the SystemView APG option: SystemView by ELANIX and Microsoft Visual C++ compiler, versions 4.0 or higher, either the Professional or the Enterprise Edition. The Standard Edition of Microsoft Visual C++ should not be used, since it is missing some of the necessary components.

From the Microsoft Visual C++ software CD, the only libraries required are the Microsoft Visual C++ Build Tools, and the Runtime Static Release Libraries. When installing the compiler, select Typical Installation, or to conserve disk space, select Custom Installation. Refer to Appendix A of this document for detailed Custom Installation instructions.
3.0 Quick Start

To create or open a complete dynamic system model in the SystemView Design Window, perform the following. (A model is complete if it executes in SystemView).

Select Tools, then Auto Program Generation (APG), and Automatic Build & Run EXE from the menu. APG will be compiled and executed, after which the results can be viewed in the SystemView Analysis Window by pressing the Analysis button on the APG toolbar.

The process can be illustrated using Costas.svu, a second order Costas demodulator for PN/PSK signals. The model is found in the SystemView example files. In SystemView, click on File, then click on Open Existing System, then Costas.sve, then click Open.

1) Launch SystemView and create or open a complete model as shown.
2) From the Tools menu, invoke the SystemView **APG build options** and select **Automatic Build & Run EXE** as shown. Clicking the right mouse button anywhere on the background of the design window can access the same command.
3) The APG executable file is built and automatically executed.

4) The button, to the right of the clock, is the SystemView *Analysis* button, and when activated, exports the results to SystemView.
5) The SystemView Analysis Window APG presents results. Notice that the plot titles are prefaced with the APG name, *Costas*. 
4.0 Creating APG Modules

4.1 APG Dynamic Link Library (DLL)

An APG direct Dynamic Link Library (DLL) is a SystemView User Code DLL that is automatically generated from a SystemView MetaSystem. It literally replaces the MetaSystem with a single User Code token, having one function with no adjustable parameters. (In the context of the model where the MetaSystem was utilized at the time of the APG creation.) Note: only connected MetaSystem inputs and outputs are translated into the User Code inputs and outputs.

The APG DLL executes faster than the MetaSystem, and the suggested use is to “freeze” finished parts of large systems while still refining the rest of the system. Another use is to accelerate some parts, while in others, retaining the option to change parameters and use Real Time Sinks and the Dynamic System Probe.

If an APG DLL is to be embedded in a larger APG, and the original MetaSystem is still available, it is more efficient to replace the User Code with the original MetaSystem before creating that larger APG module.

4.2 APG Executable (EXE)

An APG executable is a fully optimized Windows program. With minimum user interface, using a robust command loop and accelerated token calls, results in a SystemView simulation that runs much faster. Afterward, the results generated by the APG executable can be transferred to SystemView for analysis.
4.3 APG Build

When the APG option is selected from the menu, the following APG dialog window appears. Within the dialog, several APG features can be customized.

### 4.3.1 Target Processors

APG currently supports those processors that support the 32-bit Microsoft Windows operating systems.

### 4.3.2 Comments

The comments field is helpful since all APG modules look the same. The comments can be entered directly or imported from a text file.

### 4.3.3 Output File

The EXE or DLL output file can be placed in any folder available from the computer system, by clicking the Select Output File button. This invokes a standard Windows file dialog. When executed, the APG module will generate data files corresponding to each of the connected sinks in the simulation. The connected sinks in the simulation are those that are located in the same folder as the APG module.
4.3.4 APG Build
For a MetaSystem with external connections, SystemView will build a DLL. For an autonomous MetaSystem, or for an entire simulation system, SystemView builds an EXE.

4.3.5 Auto Replace MetaSystem
If checked, an APG DLL that is derived from a MetaSystem within an active simulation can immediately replace that MetaSystem token. The APG DLL can also replace the MetaSystem by disconnecting and deleting it, then inserting the corresponding APG User Code token, and connecting it the same as the MetaSystem was connected.

Note: The DLL input and output ports, must be identified by informative labels. The labeling can be done in a dialog box that appears after clicking the Label DLL I/O Ports button.

4.3.6 Finish
After the APG is defined, click the Finish button to begin the build process.
5.0 Working With an APG Executable

5.1 APG EXE Controls

An APG executable has a minimal user interface to control its execution and to possibly interact with SystemView. The following commands are provided on the APG menu:

- **Launch SystemView**: locates and launches SystemView.
- **SystemView Analysis**: exports APG data to the SystemView Analysis Window. If SystemView is not already running, it is launched first.
- **Stop**: terminates the current APG system execution.
- **Run**: begins an APG system execution or continues it, if it was paused.
- **Pause**: suspends the current APG system execution.
- **Define System Time**: this invokes the System Time Specification window as shown, with dialogs similar to that in SystemView.
- **Sinks**: a sub-menu contains commands for enabling and disabling of all APG sinks, as a group or each one individually. Sinks with a check mark are enabled, and those without it are disabled.
All of the commands, with the exception of the sink control, are also available on the APG toolbar.

SystemView APG Time Specification Window
5.2 APG EXE Execution

As APG executes, information about the system’s progress is available in the APG window:

- **Elapsed Time:** is the real time from the beginning of the execution. It excludes pauses, but does not exclude responses to the warnings, or error messages and dialogs.

- **System Time:** is the current simulation time, i.e., start time plus the current sample number multiplied by the time spacing.

- **Current Loop:** is the current simulation loop number, counting from one to the total number of loops.

- **Current Sample:** is the current iteration number within the current loop, counting from zero to the total number of iterations per loop minus one.

- **Progress Bar:** roughly indicates the completed portion of the execution, and the current iteration in relation to the total number of possible iterations, i.e., the total number of iterations per loop times the total number of loops.
5.3 APG Results
As in SystemView, the sinks in the APG EXE contain simulation results. Every sink in APG produces a SystemView plot file that is readable in the SystemView Analysis Window. Some sinks produce an additional file of a specific type and extension. Concatenating the APG name with the custom or default sink name and a specific extension creates the file names. An exception is when the External Sink file name is the same as in SystemView.

<table>
<thead>
<tr>
<th>SystemView Sink Type</th>
<th>APG Sink Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis Sink</td>
<td>Plot (PLT)</td>
</tr>
<tr>
<td>Audio Sink</td>
<td>Plot (PLT) and wave form (WAV)</td>
</tr>
<tr>
<td>Averaging Sink</td>
<td>Plot (PLT)</td>
</tr>
<tr>
<td>Current Value Sink</td>
<td>Plot (PLT) and text (TXT)</td>
</tr>
<tr>
<td>Data List Sink</td>
<td>Plot (PLT)</td>
</tr>
<tr>
<td>External Sink</td>
<td>Plot (PLT) and external sink file</td>
</tr>
<tr>
<td>Final Value Sink</td>
<td>Plot (PLT) and text (TXT)</td>
</tr>
<tr>
<td>Real Time Sink</td>
<td>Plot (PLT)</td>
</tr>
<tr>
<td>Statistics Sink</td>
<td>Plot (PLT) and text (TXT)</td>
</tr>
<tr>
<td>Stop Sink</td>
<td>Plot (PLT)</td>
</tr>
<tr>
<td>SystemView Sink</td>
<td>Plot (PLT)</td>
</tr>
</tbody>
</table>

An APG DLL, is connected within a dynamic system model in SystemView, and should only use its inputs and outputs to exchange data with SystemView. All APG DLL sinks are disabled, to avoid the potential problem of file sharing violations.

5.3.1 Importing APG Sink Data in SystemView
Importing APG data into the SystemView analysis window is accomplished by using the Import APG Sink Data button in the Analysis Window. Data from the latest run of an APG executable can be imported at any time using this feature. A single SystemView plot file can be opened in the Analysis Window using the Open SystemView Plot command under the File menu.
5.3.2 APG Sink File Management

For each sink selected in an APG system, the results are stored in separate files. After each APG execution, new results are written over the old results. APG or SystemView never delete the files. Note that the files require managing by the user.

5.4 User Code Support

User Code tokens can be incorporated in APG, using their object library files. Since the location of the necessary files to be included is unknown, this information must be supplied in a text file with the same path and name as the User Code DLL. This User Code DLL text file must have a .sva extension. On each line of that file, there must be one entry with the full path name of a User Code component, such as an object or object library file. In the simplest case it is just one object file, such as C:\SVUCODE\Release\svucode.dll and would have the extension of: C:\SVUCODE\Release\svucode.sva resulting in:

C:\SVUCODE\Release\svucode.obj

If any components have been included incorrectly, are missing, or the objects or object libraries have conflicting duplicate identifiers, a linking error occurs.
6.0 Troubleshooting

The APG log is available after every APG build. It contains the generation log, followed by the compilation log, and the linking log. With operations that are not allowed, two typical errors occur: (1) attempting to create an executable from a connected MetaSystem or (2) attempting to create a DLL from a disconnected MetaSystem.

Problems that may occur during the generation and compilation stages, and cannot be fixed on site, should be referred to Eagleware-Elanix technical support.

Problems, which occur at the linking stage, are usually related to missing object libraries and User Code object files. The problems are usually corrected by installing the missing libraries and by locating the missing files for the .sva files.
Appendix A. Microsoft Visual C++ Installation

Microsoft Visual C++ for APG must be a version 4.0 or higher, and either the Professional or Enterprise Edition, but not the Standard Edition. With the 32-bit Windows running, insert the CD into the drive and it should self-start. If not, open My Computer, select CD drive, and select Open from the file menu. In the CD-ROM folder, select and run Setup.exe. From the setup menu select Install Microsoft Visual C++.

After the Welcome, Software License Agreement, and Registration screens you will reach Installation Options. If you use only Microsoft Visual C++ for APG, select Custom to save hard disk space.

On the Microsoft Visual C++ Setup screen select only the following items:
- If using version 5.0 or 6.0, select Microsoft Developer Studio. Disregard if using version 4.0, 4.1 or 4.2.
- Select Microsoft Visual C++ Build Tools.
- Select Runtime Libraries and press Details button. On the screen that appears, select only Static Library. Press Details button again, and on the next screen select only Release Libraries. Press the OK button on this and the next screen, to return to the Microsoft Visual C++ Setup screen.

Press the Next button, and accept the defaults for the rest of the installation.

To save disk space during compiler installation, disable all options and select only Microsoft Developer Studio, Microsoft Visual C++ Build Libraries, and Static Library-Release Library. Note: Do not choose any debug libraries. The next screen that appears is for the initial keyboard configuration, select the Microsoft Developer Studio, and do not choose the old C++ 2.0 interface.
Appendix B. Custom Compilation

All of the APG generated C files and library files exist in the SystemView folder. The C source is ~apgtmp.c for both an executable and a DLL. For a DLL, the additional source is ~apgtmp1.c, and the module definition file is named ~apgtmp.def.

The two APG libraries are apglibpc.lib and apgutlpc.lib. The APG Windows interface program, and its associated resources are in apgwin.lib and apgwin.res, respectively. All of the libraries are compiled using Microsoft Visual C++ v 4.1, with 8-byte structure alignment, and are linked to the static multithreaded C runtime library. To make a custom compiler-linker call, write a batch file containing the call, name it apgbuild.bat, and place it in the SystemView folder. The batch file used by SystemView looks similar to the following:

```
cl ~apgtmp.c /nologo /MT /O2 /link /machine:ix86 /subsystem:windows apgwin.lib apgwin.res apglibpc.lib apgutlpc.lib
```

You may need to specify the full path to cl.exe, the include folder in Microsoft Visual C++, and full paths to the necessary object libraries in Microsoft Visual C++. See the Microsoft Visual C++ User’s Guide for details.
Appendix C. APG Components

SystemView components:
- SVUAPG.DLL
- APGINIPC.DLL
- APGLIBPC.LIB
- APGUTLPC.LIB
- APGWIN.LIB
- APGWIN.RES

Microsoft Visual C++ libraries:
- KERNEL32.LIB
- USER32.LIB
- GDI32.LIB
- COMDLG32.LIB
- COMCTL32.LIB
- WINMM.LIB
- ADVAPI32.LIB
- LIBCMT.LIB
- For versions 4.2, 5.0, 6.0: LIBCIMT.LIB
- For versions 4.2, 5.0, 6.0: LIBCPMT.LIB