



Gedae 5.2.1 Release Notes

May 2007

Address: Gedae, Inc.
1247 N Church St, STE 5
Moorestown, NJ 08057
Telephone: (856) 231-4458
FAX: (856) 231-1403
Internet: www.gedae.com

1 New Features

Segmented Parameters

Introduction

Primitive segmented parameter outputs allow a primitive to set the static data flow parameters to be used during the processing of a segment. They provide the benefits of static data flow (predetermined execution order and preplanned memory usage), while allowing the flexibility of dynamically changing data flow parameters on segment boundaries. Segmented parameters can be used to control otherwise static data flow parameters like `input decimate`, `output interpolate values`, and `output array dimensions`. These values can in turn affect the execution granularity, the return value of the built-in `size` and `tokensize` functions and `queue` parameters.

The segmented parameter outputs are set at the beginning of the segment and used throughout the processing of the segment. Any calculation of derived data flow parameters is done only one time at the beginning of the segment.

A primitive that has segmented parameter outputs must also set constraints on these parameters if they are connected to data flow parameters. The constraints, at a minimum, must set the maximum value of each segmented parameter used to control data flow. These constraints are used to determine memory allocation requirements on both static schedule and dynamic queue memory.

Description of Primitive Syntax

A complete example of a primitive that uses segmented parameters can be seen on the next page. Segmented parameters are declared in the primitive output section as:

```
segparam int rout;  
segparam int cout;
```

Currently segmented parameter data types are limited to scalar integer and floating point values.

The maximum value of a segmented parameter value that is used to control data flow must be specified. These specifications are called segmented parameter constraints. The constraints are only used at compile time to determine memory allocation requirements. The primitive `Apply` method must obey the constraints in the constraint section. If the primitive generates values exceeding the constraints, then it is likely that the array bounds allocated at compile time will be exceeded.

Constraints for the segmented parameters are set in the `Constraint` section of the primitive:

```
Constraint: {  
    rout <= Rmax;  
    cout <= Cmax;  
    rout*cout <= Max;  
}
```

Each constraint has a left-hand side that is a single segparam or is the multiple of segparams. Constraints with a single segparam are called simple constraints, and constraints with multiples of segparams are called joint constraints. The left-hand side of a constraint is always followed by a \leq sign. The right hand side of a constraint is any whose equation terms are input parameters and dimensions, and input and output families. At a minimum, every segparam used to control data flow must have a simple constraint.

Segmented parameters can be used in the `Start`, `Reset` and `Apply` methods like values declared in the primitive `Local` section. Like locals, the segparam values can be initialized in the `Start` and `Reset` methods and are persistent between calls to the `Apply` method.

Merely assigning the value of a segmented parameter output does not make it known to the downstream schedules. To set the value of a segmented parameter for the current segment, the `Apply` method must call the `push` function as:

```
push(rout);  
push(cout);
```

The `push` function should be called at the beginning of a segment to set the value of the segmented parameter to be used throughout the segment processing.

```

Name: seg_dfparam
Type: static
Input: {
    stream float in;
    stream int rin;
    stream int cin;
    int Rmax;
    int Cmax;
    int Max;
}

Output: {
    segmented dynamic stream float out;
    segparam int rout;
    segparam int cout;
}

Constraint: {
    rout <= Rmax;
    cout <= Cmax;
    rout*cout <= Max;
}

Reset: {
    rout = -1;
    cout = -1;
}

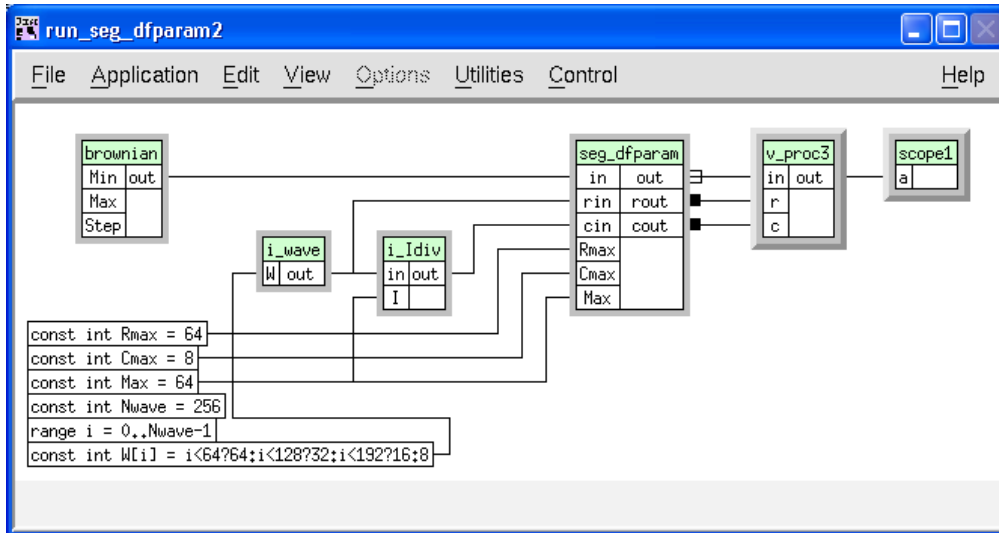
Apply: {
    int g;
    int nseg = 0;
    for (g=0; g<granularity; g++) {
        if (cout != cin[g] || rout != rin[g]) {
            if (rout != -1) {
                produce(out,nseg);
                nseg = 0;
                segment(out,SEGMENT_END);
            }
            rout = rin[g];
            cout = cin[g];
            push(rout);
            push(cout);
        }
        out[g] = in[g];
        nseg++;
    }
    produce(out,nseg);
}

```

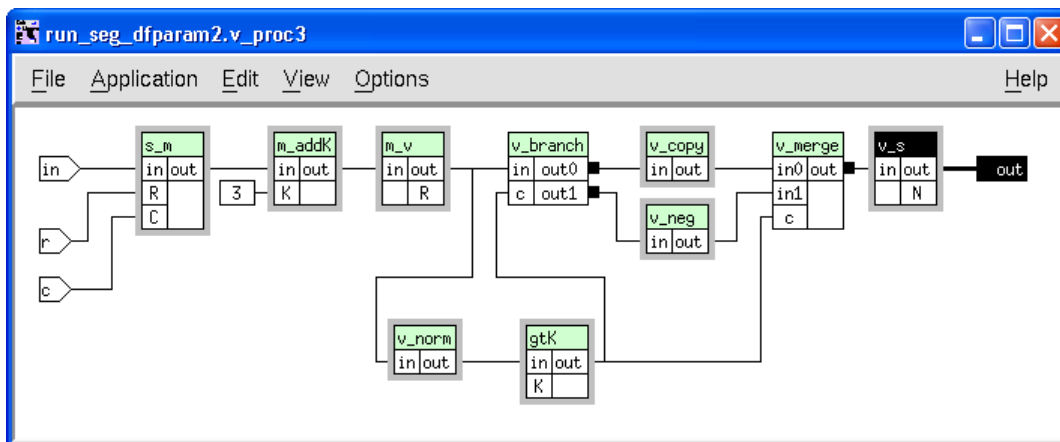
Using Segmented Parameters to do Variable Array Processing

One of the main purposes of segmented parameters is to be able to set the size of a vector or matrix to be processed during a segment; however, the segment source itself cannot be the variable vector or matrix but must be a scalar stream. The scalar stream is then converted into a vector or a matrix within the segmented subgraph using a primitive like `s_v`, `s_m`, `x_vx`, `x_mx`.

For example the `seg_dfparam` primitive above is used to pass segments to the `v_proc3` primitive as seen in the graph below.



The `v_proc3` primitive converts the scalar input stream into a matrix using the current values of the `segparams` `rout` and `cout`.

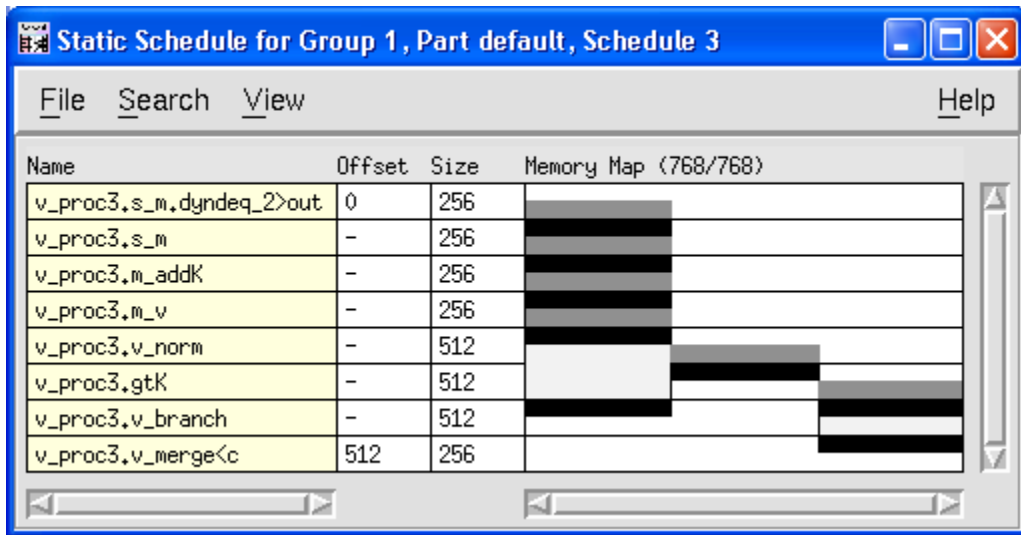


Effect of Segmented Parameter Joint Constraints on Memory Allocation

Using joint constraints of the form $rout * cout \leq Max$ can reduce the amount of memory allocated for a schedule. In the example above, plugging in the values of $R_{max} = 64$, $C_{max} = 8$ and $Max = 64$ give

```
rout <= 64
cout <= 8
rout*cout <= 64
```

Without using the joint constraint, the size that we would have to allocate for a matrix controlled by both dimensions $rout$ and $cout$ would be $64 * 8 * \text{sizeof}(\text{float}) = 2048$ bytes. Using the joint constraint of $rout * cout \leq 64$ we need allocate no more than $64 * \text{sizeof}(\text{float}) = 256$ bytes. The effect of this allocation can be seen in the memory map of the static schedule of schedule 3 below. All memory buffers in the schedule are allocated to be only 256 bytes – a savings of 8x over what would be required if the joint constraints were not used.



The screenshot shows a window titled "Static Schedule for Group 1, Part default, Schedule 3". It contains a table with columns: Name, Offset, Size, and Memory Map (768/768). The table lists several memory buffers with their respective offsets and sizes. The memory map shows the layout of these buffers in memory, with some overlapping and some non-overlapping.

Name	Offset	Size	Memory Map (768/768)
v_proc3.s_m.dyndeq_2>out	0	256	[0, 256]
v_proc3.s_m	-	256	[0, 256]
v_proc3.m_addK	-	256	[0, 256]
v_proc3.m_v	-	256	[0, 256]
v_proc3.v_norm	-	512	[0, 512]
v_proc3.gtk	-	512	[0, 512]
v_proc3.v_branch	-	512	[0, 512]
v_proc3.v_merge<c	512	256	[512, 768]

Currently, a joint constraint is only useful in limiting memory if the constraint equation explicitly has an expression of the form $R * C$ where R and C are ultimately derived to be identical to the constraint values. For example, suppose a primitive has an input declared as:

```
stream float in[R][C]
```

and an output declared as

```
stream float out[R][C+1]
```

The output size would then be given as $R * (C + 1)$. Since $R * C$ does not directly appear in the expression, the joint constraint will not be used. This failure to use the constraints supplied will be fixed in future releases.

Limitations

The following limitations currently apply to the use of segmented parameters.

1. All data flow expressions using segmented parameters must be monotonic increasing functions. Expressions like R/C where C is connected to or derived from a segmented parameter will result in too little memory being allocated.
2. Segmented parameter outputs from the controlling primitive cannot be connected to graph parameters. If a derived parameter is needed, then it must be calculated in the segment controller primitive.
3. Segmented parameters must be scalar values of type int.
4. Delay and Overlap cannot be controlled dynamically.
5. Even fixed delay and overlap cannot be of tokens of a dynamic size. For example, the declaration

```
stream float out[R][C] (1,0,1)
```

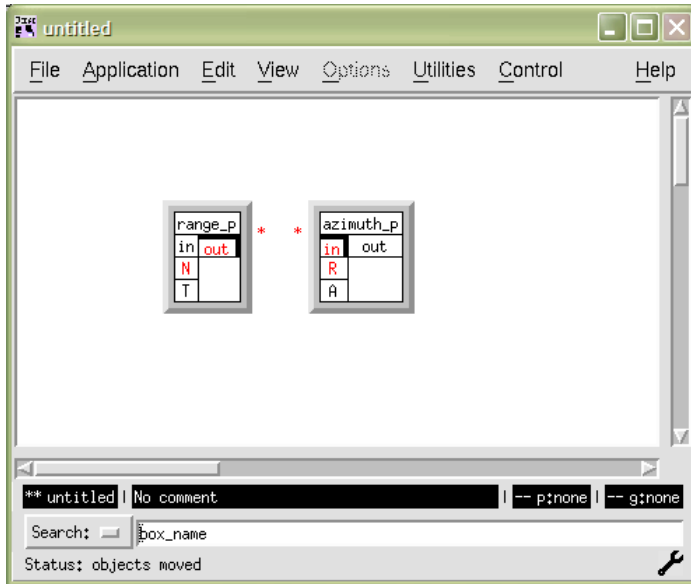
is illegal if either R or C is a dynamic data flow param.

6. A dynamic queue that contains matrix token types will allocate space for the tokens that are $R_{max} * C_{max}$ even though $R * C$ may be constrained to be less. The queue space will be larger than it needs to be.
7. Derived data flow expressions that contain both terms of a joint constraint but do not explicitly have the multiple of the two-segmented parameters included in the joint constraint will not take advantage of the constraint.

Gedae Editor Extensions

Constants Table

The Constants Table displays all constant data in the application and allows the developer to set any settable constants. A settable constant is currently defined as a constant input to a subgraph, which does not have a source. The table is launched by the View->Constants Table menu item. The following graph has two settable constants, the N input to range_p and the R input to azimuth_p.

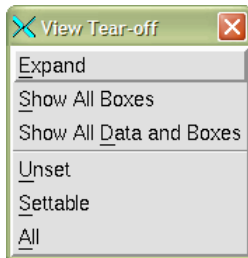


The developer does not have to attach to application to launch the Constants Table. The Constants Table for this graph is shown below. The values of all externally set constants are displayed with a gray background, as they cannot be set through this table (however, the menu item Options->Edit Data... is available for directly altering the data item). The values of all settable constants are displayed with a white background.

The screenshot shows a window titled 'welcome Constants' with a menu bar (File, Edit, View, Options). The main area contains a table with two columns: 'Name' and 'Value'.

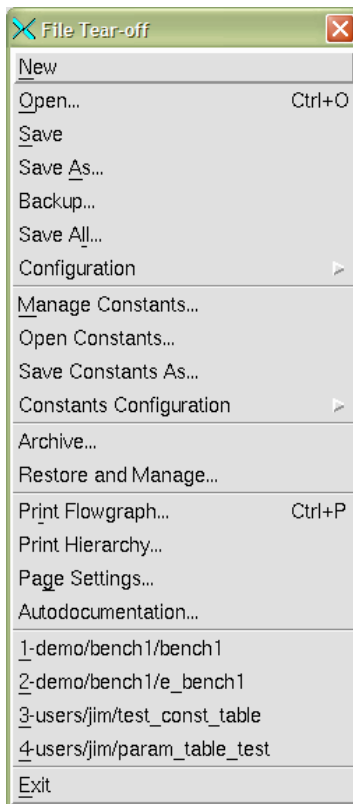
Name	Value
int R	unset
azimuth_p	
int Rm1	R-1
int one	1
int zero	0
int N	unset
range_p	
int M	1<<n
int M1	1<<n1
int Pad	M1-N
int n	log(N)/log(2.0)
int n1	M<N?n+1:n

The View menu is able to filter the table based on the type of constant. View->All shows all constants. View->Settable shows all settable constants. View->Unset shows all settable constants that are not currently set.

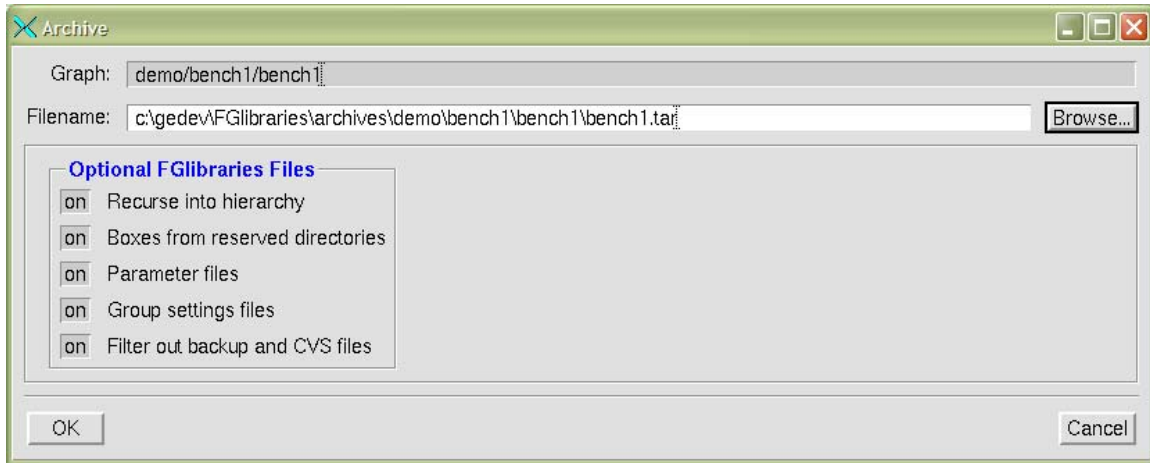


Archiving From the Flow Graph Editor

The Flow Graph Utility (FGU) capabilities of creating, managing, and restoring an archive file have been incorporated into the Flow Graph Editor (FGED). The File menu in the FGED has new items Archive... and Restore and Manage...



The File->Archive... menu item allows the developer to create an archive from the flow graph. The archive will be saved in the location specified beside Filename. This location can be edited directly, or a new location can be found by using the file dialog launched by the Browse... button. When creating an archive, there are five options, as shown in the dialog below.



These options are

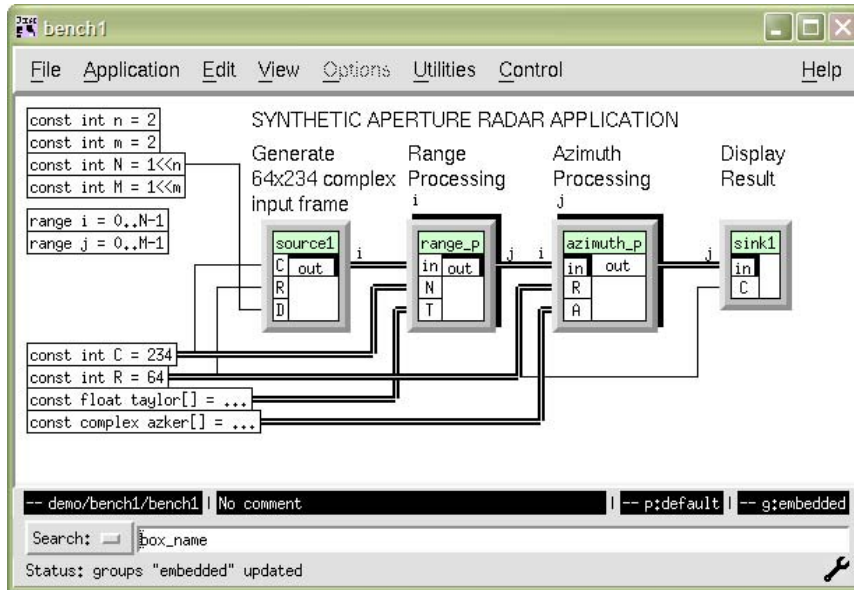
- Recurse into hierarchy – include all primitive and flow graph files used to create the application, not just the top-level flow graph.
- Boxes from reserved directories – include boxes from directories that are shipped with Gedae.
- Parameter files – include all parameter files for the application from the parameters directory.
- Group settings files – include all group settings files for the application from the group settings directory.
- Filter out backup and CVS files – filter out parameter and group settings files that end in ~ or .bak as well as the CVS directory.

The File->Restore and Manage... menu item launches the same archive browser dialog available in FGU. A new archive browser will be available in a future release.

Equivalent to FGU, the archives used are TAR files that can be used externally to Gedae using the tar utility, WinZip, and a variety of other programs. Note, the FGU executable is still made and shipped with Gedae and is available in the \$GEDAE/<os>/tools directory.

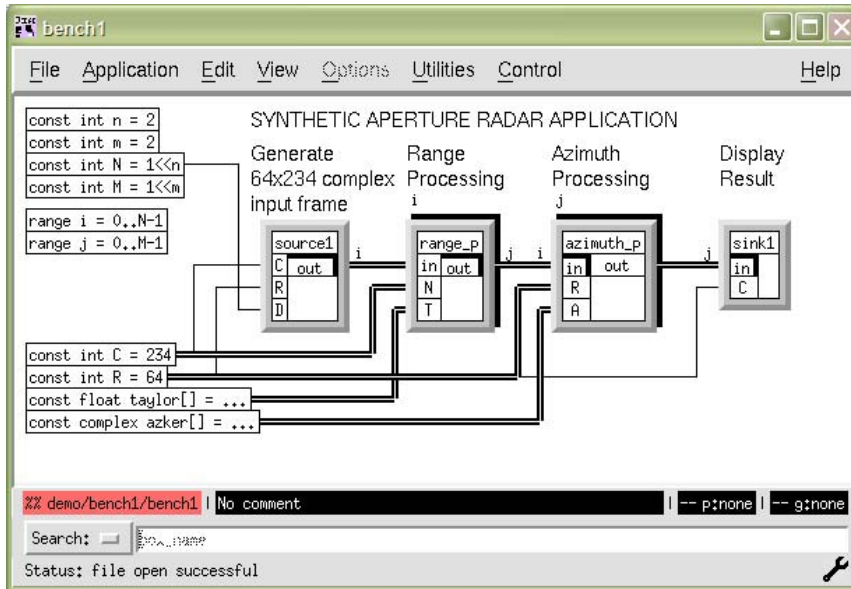
Status of Parameters and Group Settings

Gedae is more observant of the status of parameters and group settings. The flow graph file information bar (the bar colored black on the Flow Graph Editor) now presents to the user which parameter and group settings files are loaded. The section labeled “p:” marks the parameters file and the section labeled “g:” marks the group settings file. Additionally, Gedae observes when these settings have changed, marking “***” in the file information bar if the files are dirty and “—” if they are not dirty. Upon exiting or killing a flow graph, Gedae will prompt the user to save any dirty application files.

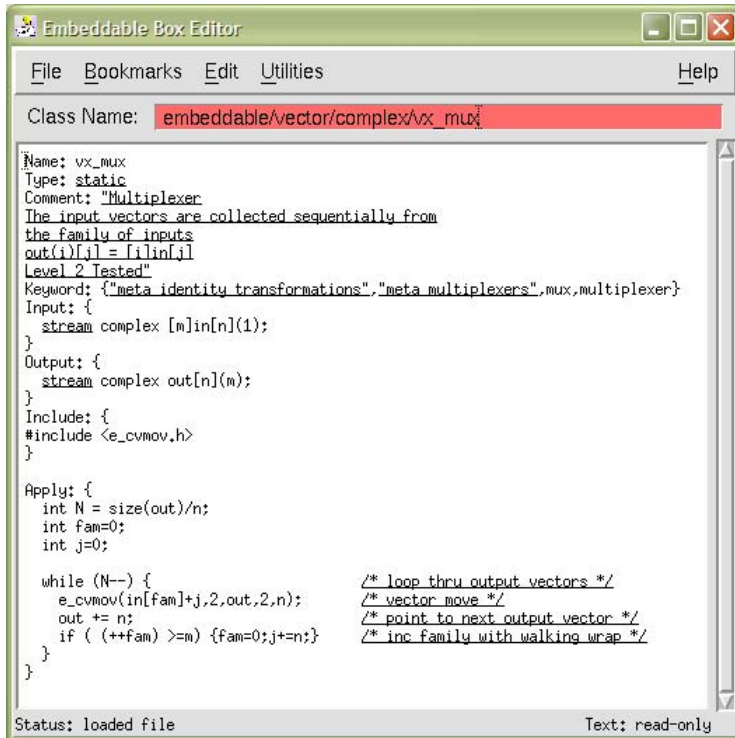


Read-Only Files

Geda is more observant of the read-only settings of files and will disallow editing of read-only flow graphs and primitives. If a file is made read-only, the file information bar is colored red both on a flow graph



and on a primitive.



In the flow graph file information bar, the symbol “%%” also identifies the file as being read-only.

The Hierarchy Table, launched via Application->Hierarchy, also identifies the read-only status of all box files. Additionally, the Hierarchy Table has menu items Options->Recursive Read-Only and Options->Recursive Writeable to set the permissions for all box files in selected subgraphs, as well as, Edit->All Read-Only and Edit->All Writeable to set the permissions for all box files in the flow graph.

The following new menu items are available in the Flow Graph Editor to change the read-only status of the corresponding types of files.

- File->Configuration->Change Permissions...
- File->Constants Configuration->Constants Permissions...
- Application->Parameters Configuration->Parameters Permissions...
- Application->Group Settings Configuration->Group Settings Permissions...

Note that the constants, parameters, and group settings menu items bring up a file selection dialog. The File->Change Permissions... menu item is available in the Primitive Editor to change the read-only status of a primitive file.

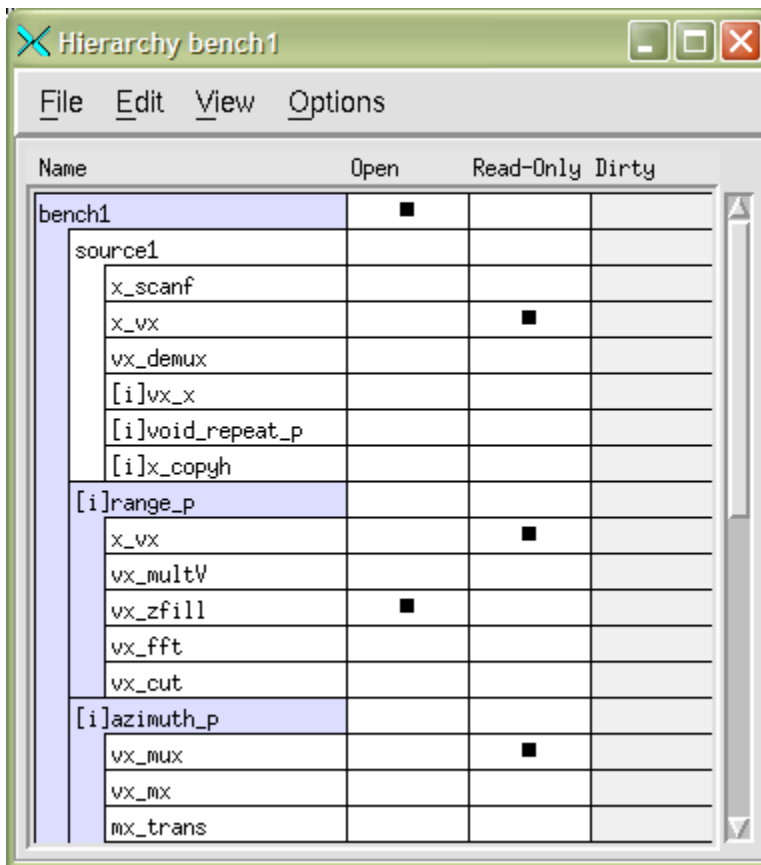
Additionally, a new configuration management (CM) menu item has been added. Many CM tools require issuing a CM command to begin editing a file. Gedae's default CM support is for CVS, and in CVS this command is called edit. The following menu items are available to access the “cvs edit” command:

File->Configuration->Edit
 File->Constants Configuration->Edit Constants
 Application->Parameters Configuration->Edit Parameters
 Application->Group Settings Configuration->Edit Group Settings

The File->Edit menu item is available in the Primitive Editor to issue this command on a primitive file. The syntax of the call to the underlying CM command can be customized through the Utilities->Configure Version Control... menu item.

Flow Graph Diff

Flow graphs can be diff'ed with FGU archives in the Hierarchy Table, launched via Application->Hierarchy. The menu item "Options->Diff With Archive" brings up a file selection dialog to select the archive with which the graph will be diff'ed. The differences are shown by color-coding the name entries in the Hierarchy Table.



The color-coding is

- Red – Data flow different; for example, a box has been added.
- Orange – Functionally different; for example, a method has been changed.
- Yellow – Data different; for example, non-data flow parameters have been changed.

- Green – No difference.
- Blue – Cosmetically different; for example, a vertex has been moved.
- Violet – Pervasively cosmetically different; for example, the name of a box or data has been changed.

These difference types can be queried by selecting Options->View Diff Type. After diff'ing with an archive, the user can view a text-based diff of individual files using Options->View Diff Results. This text-based diff uses diff programs built-in to the operating system (diff on Unix, windiff on Windows). To change the program used to do the text-based diff, set the \$DIFF environment variable.

2 Bugs Fixed

SCR Number	Title	Problem Description
002331	Get configuration control status of all boxes in graph	It is now possible from the File menu to get the configuration control status of all boxes in a graph.
002388	Prevent recursive calling of X event handler	Recursive calls to the X Event Handler function need to be prevented. Such recursive calls have been observed when setting the priority of a primitive from the Firing Granularity Table after the Gedae graph has been run. Preventing the recursive call has avoided this problem.
002414	Subscheduled Sync Box with dynamic input does not work	If a primitive with a parameter connected to an encoded queue is subscheduled, and if it has a dynamic input, then when the primitive is waiting for data on that dynamic input it will erroneously go to the "retry" state rather than the "queue-wait" state. This causes the primitive to appear to succeed on the second firing even though no new data arrived.
002424	Open syntax errors in \$EDITOR returned to release	In Gedae 4.9, on a file open, a text editor would be brought up automatically if there were any syntax errors. This feature was unintentionally disabled in Gedae 5.0. Additionally, the old capability would sometimes bring up the parent graph of the bad file instead of the bad file (which would allow the user to circumvent the error but not to fix it). Now the capability always brings up the file with the syntax error.
002428	Segfault on Arc Display when >1 instances of box in parent and only one open	Two instances of subgraph "B" are in graph "A". Open one of the "B"s and then display the connections of a family arc in the arc display by double clicking on the arc. If that does not segfault, retry with the other "B" and it will segfault. Code was using first found instance of "B" to find editor structure, and we cannot guarantee that all boxes have editors associated with them.
002433	E function e_dvrfft does not work	The E function e_dvrfft cannot be used because there is no corresponding e_drfftws function to set up the weights buffer.
002434	AFGexit error - embPause called twice	Calling AFGexit from distributed graph can cause Gedae to fail with the message: Error in box E_vmov.v_Compare_Exit.v_cmp, Schedule 1 Partition default, Schedule 1: Pause called while already paused.
002436	Segfault in Trace Table when external state primitives are in graph	When bringing up the Trace Table a segfault can occur if there are boxes of type "state" in the graph.

002441	e_dctwts.c is incorrect	The function e_set_cdtwts is spelled incorrectly in e_dctwts.c It should be spelled e_set_dctwts.
002442	Deleting parameter arc with fan-out causes segfault	Deleting an arc on the graph that has multiple parameter destinations can cause Gedae to segfault. For example, an arc that feeds a subgraph input that fans-out to multiple destination parameters could cause the problem.
002443	Segfault when deleting box in partially attached graph	If a graph has been "attached" but a box is deleted from the graph that has not attached, then the graph can segfault. This might be caused if the box has compilation errors and was added after the graph was successfully attached.
002444	Destroy callback error in widget/Xm/Text primitive	The widget/Xm/Text primitive failed to set the widget state variable to 0 in its destroy callback. This differs from all the other primitives that have destroy callbacks in the widget/Xm directory and probably accounts for a segfault observed in the Text primitives destroy callback.
002446	Improve performance of e_vabs	Change call in e_vabs from fabs (double version) to fabsf (float version).
002453	Remove old NT audio boxes and require AFG for NT audio	
002454	Erroneous copy boxes added at segmentation output boundaries	At segmentation output boundaries, erroneous copy boxes can be added to fix inplace problems. The copy boxes are added between boxes in different schedules, which is the source of the problem. In some circumstances this error can cause a segfault.
002459	CodeGen Fails on TigerSharc when a Partition is mapped to the Command Program Host	A segfault occurs during the Code Generation phase of compiling a graph to a launch package if one partition is mapped to a TigerSharc and another partition is mapped to a target host process.
002461	Do not trace monitor events in GSIM	If a GSIM graph deadlocks, the trace buffer gets filled with monitor events. These events are now filtered out when running a graph under GSIM.
002464	Dynamic output to DSA and inplace box on same proc assert error	An assert error occurs if a dynamic output is connected to an inplace destination on the same processor and a second destination on a different processor and the transfer mechanism type selected is a DSA type. This problem has been fixed.

002466	Gsim segfaults when no threads are runnable	Gsim segfaults when no threads are runnable.
002469	File->New can cause segfault	After a graph has already been opened in the Gedae Editor, clicking File->New can cause a segfault.
002470	Make it possible for subschedule inputs and outputs to be inplace	The algorithm for determining if subschedule inputs and outputs should be inplace has been improved and corrected.
002474	Eredhat/Esolaris fail when user directory is underneath system directory	If the admin installs the Gedae user directory underneath the Gedae system directory, a bug introduced in 5.1 will prevent launch packages from being fully linked. While codegen'ing the link step, if the system directory is in the path, Gedae swaps the string with a Windows-style environment variable. The Unix shell does not understand this variable. As a workaround, users should simply move their user directory from outside the system directory's branch of the file system.
002477	Bringing up launch package dialog causes unpredictable crashes	Bringing up a launch package dialog frees the cons cells in an applications parameter list. This allows these cells to be used for other things and causes the application to crash at unpredictable places.
002480	Do not allow editing of primitives while graph is running or continuable.	It should not be possible to edit a primitive while a Gedae graph is running. This restriction makes the primitive editing policy consistent with the graph editing policy and avoids a segfault that occurs due to the group state erroneously changing (while the graph is running) when the primitive is saved.
002493	Compiling target command program should fail if hostCompileLibs does not exist.	If the hostCompileLibs does not exist for a target host, then the compilation will procede and either fail at a later point with undefined symbols or in the case of VxWorks will succeed and fail at runtime. The build system has been changed to detect the nonexistence of the hostCompileLibs, print an error message to the terminal, and abort the compilation at that point.
002494	Segfault when setting firing granularity after graph runs	
002510	The graph drawing area is badly sized if opening small graph after large graph.	If a graph with a small drawing area is opened after a graph with a large drawing area, then the editor drawing area is still large after the small graph is opened. This commonly results in scroll bars being used on graphs, which would not require scroll bars if the drawing area was properly sized.

002514	Search Dialog does not work on Solaris	Search dialog will not display in Solaris. The \$GEDAE/tools/searchData.mkw datafile is an integer binary file and must be byte swapped to run properly on Solaris. As a workaround, existing users can use www.gedae.com/searchData.mkw or byte swap the file themselves.
002515	Stop followed by Terminate produces erroneous error message.	Control->Stop followed by Control->Terminate prints the following erroneous error to the terminal: ERROR: appStop: Cannot stop an application that is not running.
002519	Segfault after Run, Terminate, Delete primitive, Run when there are multiple groups.	Assume a graph has two groups, A and B. The Gedae Development Environment can segfault after the following sequence. Control->Run Control->Terminate Delete a primitive from Group A Control->Run Then Group B can segfault during allocation phase. The problem is that in this scenario a group can be rescheduled without having first freed its old schedules. These schedules are remembered and operated on although they are invalid.
002525	Dimension mismatch between parameter and steam inputs not reported	Parameter dimensions that have the same name as stream dimensions are not checked for consistency.
002526	Cursor does not change if parent editor is not open.	If the parent/ancestor editors are not open, then the cursor will not change or initialize.
002527	I/O cannot be changed if parent editor has not been opened.	If the parent editor has not been opened, then the Edit Data dialog will not accept changes to I/O or report the problem.
002531	In FGTable, if enter/down/up causes a scroll, then there can be a segfault.	Pressing enter in a text entry cell at the bottom of the window causes the FGTable to scroll down to reveal the next valid cell. The selection of the next valid cell was not done correctly and often caused a segfault.
002532	e_rect does not work inplace	When this function is called inplace, as happens within the vx_pol2rec primitive, the line: <code>c->re = a->re*(float)cos(a->im);</code> overwrites the input <code>a->re</code> so that the following line: <code>c->im = a->re*(float)sin(a->im);</code> produces an incorrect result.
002537	Parameter table lets you change data flow parameters when running.	The user should not be able to change data flow parameters in the parameter table while running.
002541	Parameter table segfaults if float is entered into integer cell.	

002547	Premature unlocking of blocked schedule in merge tree following distribution.	If a schedule that is controlled by an exclusive lock is blocked, if the schedule is the merge part of the exclusion and if it follows distributed subgraphs, then it is possible that the schedule will incorrectly release its lock when an end-of-segment is propagated up to it.
002548	Cannot increase queue size of queue whose memory is shared by exclusive subgraphs	If Gedae automatically tries to increase the queue size of a queue that shares its memory with other exclusive segments, then Gedae segfaults during the freeing of the original memory.

3 Known Bugs

SCR Number	Title	Problem Description
002012	Running two VxWorks processes on the same processor.	This problem occurs when trying to run two separate Gedae generated VxWorks executables on the same processor; however, the entry point for each executable has the same name, VxWorks_main, making this impossible.
002015	Multiple exclusive sources with some of the sources not used by every mode.	If there are multiple exclusive sources to a family of modes and some of the sources are not used by every mode, then Gedae crashes during development time scheduling. For example, if two exclusive branches drive three downstream modes and one of the branches has one of its outputs unused by the third mode, then this causes a segfault during scheduling. The workaround is to add dummy inputs to the modes to allow all the sources to be used by every mode.
002019	Graphs with host to target control ports fail on Linux and Solaris	The problem is that the host is not performing mailbox services while it is waiting to establish a control port to the target processors. Unix processors require the host to perform these services in order to make connections.
002020	Check in all licenses on exit.	Gedae relies on FLEXlm to find checked out licenses that are no longer in use and check them back in. This works on most systems, but a BSP user reports that it does not work on their system. We should explicitly check in all licenses on Gedae exit.
002022	Inplace box scheduling problems.	Copy boxes occasionally need to be added to a graph by the user to avoid scheduling problems associated with primitive outputs marked as being inplace with an input.
002024	Connecting Graph Parameter to User Define Type Segfaults.	Gedae erroneously allows standard C parameter types to be connected to user defined parameter types. This type of connection causes Gedae to segfault.
002047	Allow Changing Length of Parameter String at Runtime.	Changing the value of a string parameter to a stream may cause Gedae to stop executing. Gedae stops executing if the length of the string is changed.
002054	Modifying a running graph segfaults.	Modifying a Gedae graph that is running can cause a segfault. This problem has been reported several times but has not been duplicated by the Gedae support group. Most edits are disabled during graph execution.

002062	Outlaw segmented static schedules controlled by nondet inputs with multiple boxes.	Segmented static schedules controlled exclusively by nondet inputs and that contain more than one box should be outlawed. These graphs are currently considered problematic and can produce unexpected results.
002064	Outlaw pointer streams followed by delays or overlap	Pointer streams followed by delay boxes or boxes with input overlap parameters do not work and should be outlawed.
002065	Pointer stream followed by an Inplace box	A pointer stream cannot be followed by an inplace box that has an Apply method because the execution of the box will modify the pointer's value. A workaround is to add a copy box between the pointer and the inplace box.
002078	Gedae can go to sleep if processes are polling.	A Gedae process can go to sleep if a process is polling. The sleeping is only seen on the NT BSP, which is currently the only BSP that implements the sleep capability. Gedae should only sleep when all schedules are in the paused state.
002104	Handling of null segments in distributed graphs	If a segmenter controls a segmented subgraph that is distributed and if the segmenter produces null segments, then parts of the distributed graph that are not directly controlled by queues will not see the end-of-segment.
002158	Trace Table send/recv webs do not work from attached launch package	When the Gedae Development Environment attaches to a launch package, the Trace Table send/recv webs do not work. No send/recv webs are displayed.
002221	Search on type does not follow route boxes.	If you search on the type of the input to a scope1, it does not recognize the type because there is a route box between the input and the next primitive.
002244	Using f6 to disconnect a constant can cause a segfault.	Disconnecting a constant source from a constant destination using the f6 button causes a segfault. This occurs because a stub is now connected to the constant destination and the evaluation of the constant fails.
002245	Gedae does not prompt the user to save a graph.	If a graph has been modified using the f6 cursor and that was the only modification, then Gedae does not prompt the user to save the graph when exiting Gedae.
002261	embTerminateError called from Reset does not stop the primitive.	If embTerminateNormal or embTerminateError are called from a Reset method of a graph that has more than one static schedule, then the static schedule containing the primitive is not terminated, as it should be. The schedule Apply methods will be called even though one of the schedule's Reset failed.

002267	Nonfamily output connected to family input causes graph to crash.	If a nonfamily stream source is connected to a family stream input, then Gedae segfaults when the user runs the graph. This type of graph should be detected as an error, and Gedae should not be allowed to run.
002282	Limitations on the use of typedef primitives	Typedef primitives will not work if connected to variable vectors or matrices. Typedef primitives will not work if connected to route boxes.
002302	MCP3 Parameter Support	MCP3 BSP does not yet support parameters.
002303	MCP3 Granularity Support	MCP3 BSP does not yet support granularity for FPGA->PPC comm.
002312	The Gedae dy4av2 BSP function e_zvrcip can kill process.	Calling the Gedae Dy4av2 BSP function e_zvrcip can cause the process to exit with an arithmetic exception. The problem is that even if the argument to e_zvrcip has only its real or imaginary part set to zero (but not both), then a divide by zero error still occurs.
002349	FGU does not work with long pathnames	When moving a FGU tar from Solaris to Windows, the Windows system cannot reproduce long filenames. This seems to be a problem with the Windows tar executable.
002360	Trigger boxes with Reset method but no input parameters do not get fired	A trigger box that has a Reset method but no other inputs does not get fired when the user selects Control->Run.
002361	Eval boxes with no inputs are not included in the launch package.	This feature can be confusing as when the Eval box calls getcwd. In this case the launch package will use the wd from which it was created - not the one it is run from.
002378	Create Subgraph does not support multidimensional families	Create Subgraph cannot handle boxes and data that use multidimensional families.
002399	Gedae built-in function names should be ignored if used in nonfunction context.	Gedae built-in functions like "time" should be ignored if not used as a function. That is time(x) should be converted by the Gedae parser but x.time (or even x.time(x)) should not.
002400	Family of array parameters to a trigger box does not codegen correctly.	An input parameter like float [N]in[Max] does not codegen correctly for trigger boxes. The data values are not correct. If only one family member is set, then the box still executes; however, the array dimensions are invalid.
002406	Primitive with EndOfSegment but no Apply does not work.	If a primitive has an EndOfSegment method but no Apply method, the primitive is not included in the running application and the EndOfSegment method is never called.

002419	FGTable edit box does not recognize termination.	Gedae does not accept some FGTable entry while the graph is running. If you are trying to edit an FGTable entry, then terminate the graph so that you can change the value, the FGTables do not automatically recognize the graph has terminated. The user must unselect the entry area then reselect it to enter the data.
002420	Microphones sporadically stop collecting data on AFG for Linux.	AFG would sometimes stop getting data from the microphones. Users were successfully running the graph for several tries, but at the next execution, it would stop working. AFG recognizes the microphone but just does not produce any data.
002423	Dy4Av2 Installation Requirement	We needed to remove the files rsh.exe and cygwin1.dll from %gedae%\nt\bin because delivering these files and placing %gedae%\nt\bin in the users PATH variable caused many users to have conflicting versions of cygwin1.dll in their path. It is now necessary to copy cygwin1.dll and rsh.exe from the cygwin bin directory into %gedae%\nt\bin. For example, these files can be copied as: copy c:\cygwin\bin\cygwin1.dll %gedae%\nt\bin copy c:\cygwin\bin\rsh.exe %gedae%\nt\bin
002431	Local dcomplex variable is treated like a complex variable.	If you create a dcomplex local Local: { dcomplex temp[N]; } and then look at the Memory Map, the Memory Info dialog states Type: complex ... Elem Size: 8 ... The correct Elem Size is 16. Running with a local dcomplex causes a segfault because this buffer is not big enough.
002432	Connection from float on canvas to double input parameter does not cast correctly.	If there is a double parameter input to a primitive and a float on the canvas, the user can connect the float to the double, however the cast is not performed correctly. The value inside the primitive is 0. Either cast floats to doubles correctly or disallow the connection.
002447	Entering Launch Package Directory in launch info dialog does not cause recompile.	Entering a directory name in the launch package creation dialog does not cause a group that is in the compiled state to move down to the allocated state. As a result, hitting the Make button on the launch package dialog has no effect.
002450	Loading group settings when the target host is set can fail.	If group settings are already loaded in which the Target Host is set to something other than default and if a new group setting is loaded in which a partition is mapped to a target logical number that appears in the Target Host configuration file but not the default configuration file then Gedae complains that "Logical processor 100 is not in embedded config file" and fails to set the "Run on Target" toggle on.

002451	Undefined symbols of form I0023_recv in target executables.	The error message: "Warning: lookupDirectorySerialNumber: directory name %s not in database and database is locked\n" can occur when compiling a distributed graph mapped to a target processor. This error may cause the undefined symbols of the form I0023_recv when linking the target executables.
002467	Symbol "hpeerror" multiply defined in Exceed and Visual C++	Including both <X11/X.h> and <stdlib.h> introduces a compiler warning for some mixtures of versions of Exceed XDK and Visual C++. The problem was noticed when building with .NET 2003, and an evaluation version the eval version of XDK 2007. This problem is not noticed with the same Visual C++ and the full release version of XDK 2007. A workaround is to add a line like #define hpeerror hpeerror2 in front of the include for the 2nd of these 2 files.
002472	The build in size function does not work for dynamic inputs	If a primitive input is declared as dynamic stream in; then the build in function size does not work correctly. The value of size(in) will be zero instead of the number of data elements in the queue. The primitive should use granularity instead. If the declaration is dynamic stream in[N][M](D), then the Apply method should replace size(in) with granularity*N*M*D. This problem should either be detected or fixed.
002473	Gsim exits if group is set to run on target model and there is no target model.	Should not allow graph to run in such a situation.
002478	Memory Map Dialog "View->Changes" does not work for dev env talking to target command program.	In the situation where the development environment is talking to a target processor through a target command program, selecting the View->Changes option on the Memory Map Dialog for the target causes Gedae to segfault.
002479	If a compile is not necessary – do not say complete and successful - say up to date.	If a compile is not necessary – do not say complete and successful - say up to date.
002498	Partition not set correctly when adding new box to partitioned subgraph	When adding a box to a subgraph that has its partition set in the Partition Table, the new box should have its partition set to the subgraph partition. This does not happen. A workaround is to save the group settings (if needed) before adding the box and then reload the group settings after the box is added.
002499	Changing connection does not mark graph as dirty.	Linux PPC
002502	Multiple dyndeqs are added when fanning out from dynamic output to multiple static dests.	

002503	Variable vector dyndeq's are inefficient.	
002507	Hitting return in blank Launch Package Dialog directory field.	If a return is hit in the Launch Package Dialog's directory field and the field is blank it does not "unset" the directory value as it should, but rather sets the directory value to an empty string.
002521	Don't allow setting Firing Table Granularity if Gran Mult > 1	When using the Firing Table, if the Gran Mult field on a schedule is > 1, then the user should not be allowed to set the Granularity field of any primitive in the schedule. Additionally, setting the Gran Mult should clear all the schedule Granularities.
002549	Memory leak when increasing the queue size of a shared queue.	When increasing the queue size of a shared queue. Gedae does not free the original memory. This results in a memory leak. If the queue size is increased a second time, it would be possible to free the memory allocated the first time. Gedae does not keep track of this.
002550	FGU of hierarchical typedef boxes	FGU does not transfer hierarchical typedef boxes correctly. The typedef used to define the input of the box is set to the old directory rather than the new.
002551	Graph Stalls	A rare condition can cause a graph to stall (or segfault) when the controlled static schedule is partitioned to two processors in the following form: A->B->A. The problem scenario is that the schedule is partitioned into three parts, with the first and last parts mapped to the same processor. Usually Gedae puts the parts mapped to processor A in the same static schedule; however, to allow efficient pipelining, Gedae splits the two parts mapped to processor A into two different static schedules. They are numbered n.1 and n.2 (for example 2.1 and 2.2). To see if any schedules have been broken into two parts, the user can pop up the Schedule Info Dialog and see if any of the schedule names contain a decimal point. The decimal point in the schedule name does not necessarily indicate a problem. The problem only occurs when the data source driving the processing is faster than the graph, causing the control message queue to back up and overflow. The condition is rare because the problem only happens when the graph is not keeping up with the input data rates.
002553	Embedded build can require a makeGEDAE CLEAN	If an application is repartitioned, then the target executables do not get relinked. The problem is that all the .o files are older than the targets, and the fact that there is a new link line does not force the target library and target executables to rebuild.
002554	Unterminated comments	Unterminated comments cause the Gedae parser to segfault.
002555	Parser problem	The Gedae parser does not handle an odd number of quotes well.

002556	Arrays of strings not allowed	Gedae currently allows string array graph parameters to be declared as: <code>const string X[] = { hello , world }</code> or <code>string X[i] = [i]Y</code> where Y is a family of strings. In either case, the values so declared are not correctly set, and therefore, should be considered illegal.
002557	Function <code>appFree</code> memory leak	A command program running on VxWorks does not free all the resources allocated (memory, sockets, etc). The <code>appFree</code> function must release everything allocated. Gedae should automatically generate a call to <code>appFree</code> for the standard <code>exec-host</code> command program.
002558	External Code does not recompile	<code>Make</code> is not called after a successful run, so changes to code listed in the <code>Personal_Emb_Obj_List</code> do not get recompiled. To force the recompile, it is currently necessary to change something from the Gedae GUI. For example, saving a primitive or toggling the Group "Run on Embedded" toggle off and on will force a recompile.
002559	Large Graphs fail to display on flattened graph.	If a graph is too large, then it cannot be displayed on the flattened graph. This occurs when the flattened width or height exceeds the allowable pixmap width or height of 32768.
002560	Primitive cannot recompile	If a primitive Input, Output or Local section is modified at runtime, then Gedae segfaults when the primitive is recompiled, and the graph is rerun. Currently, the user must exit Gedae after a primitive Input, Output or Local section has been modified.
002561	DSA with fan-out does not work for some BSPs.	If a box output fans-out to several boxes mapped to several different processors, then the DSA communication mechanism does not work correctly for Mercury and Sky BSPs.
002562	FFT primitives only work with power of 2 sized vectors.	The FFT boxes do not support non-power-of-2 lengths; however, the comments make no mention of this fact. If these boxes only support a power-of-2, then it would be useful to have a separate set of boxes that support a non-power-of-2.
002563	Constants propagated through typedef boxes	Constants propagated through typedef boxes cannot be used for instantiation.
002564	Stream box with push in hostless launch package.	If a stream box contains a call to push and it is made part of a hostless launch package, then the launch package will fail to compile, as the code for the push is not included in the standalone library.

002565	Transferring doubles between host and target with different endian (Dy4av2)	Currently, the Gedae BSP does not support providing byte swapping of doubles when transferred between the host and target processors.
--------	---	---