



# Gedae 5.0.1 Release Notes

September 2006

Address: Gedae, Inc.  
1247 N Church St, STE 5  
Moorestown, NJ 08057

Telephone: (856) 231-4458  
FAX: (856) 231-1403  
Internet: [www.gedae.com](http://www.gedae.com)

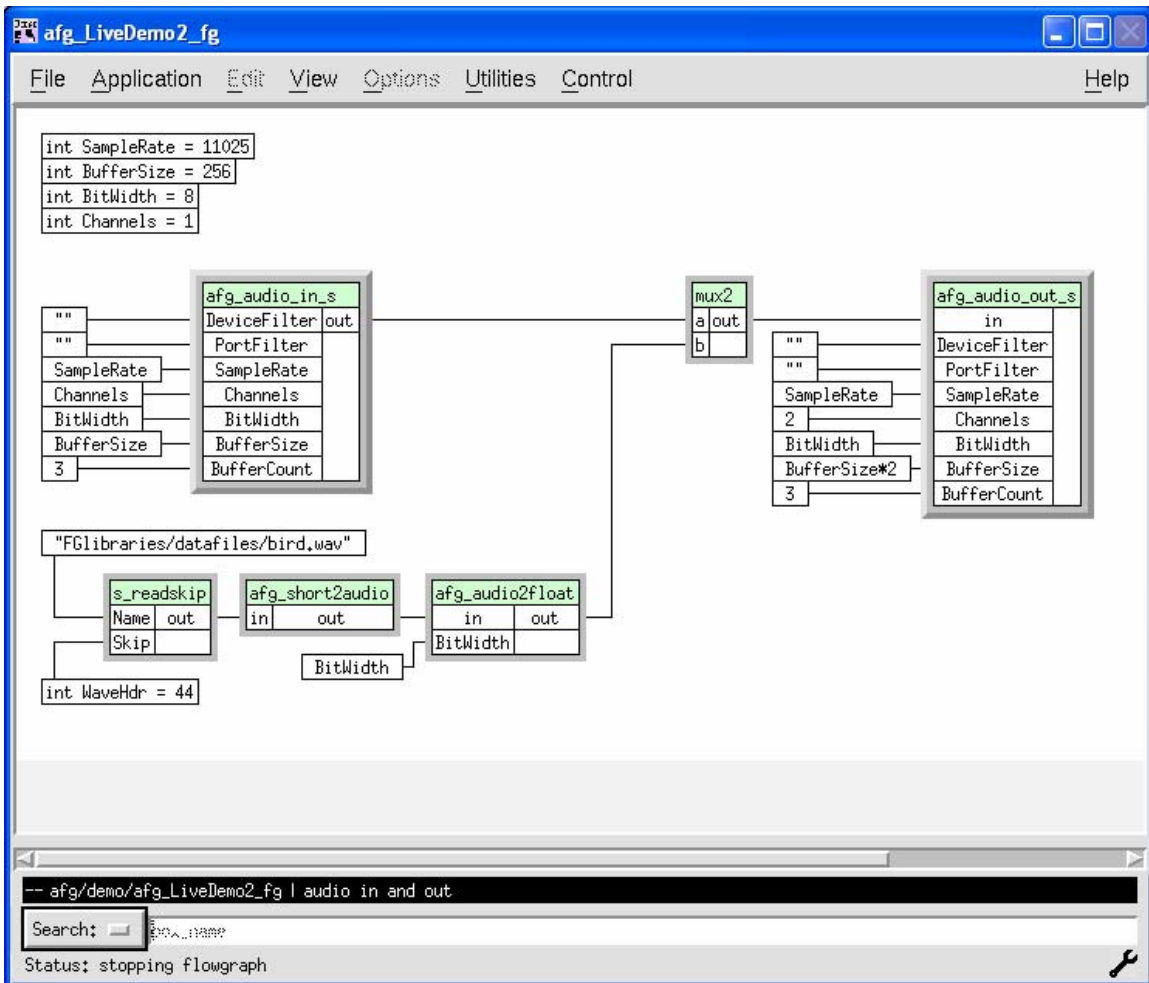
# 1 New Features

## Audio for Gedae (AFG)

A new set of primitive functions has been added to support audio processing in Gedae. Primitives include audio input and audio output. The primitives are written using a new API that is ported to both Linux and Windows platforms. AFG can be downloaded from

<http://gedae.com/downloads/releases.html>

Below is a graph that mixes the data from a microphone input and the data from a file and plays it out to the audio output device which accepts the 2 channel input.



Follow the installation instructions in the AFG\_installation.txt file in the downloaded folder.

## Video for Geda (VFG)

A new set of primitive functions has been added to support video processing in Geda. Primitives include videos input and video output. The primitives are written using a new API that is ported to both Linux and Windows platforms. Below is a simple Geda graph showing a video input device connected to a video output device.

Valid Modes are:

- yuv422
- yuv411
- yuv420
- yuv410

Not all devices support all these modes.

Valid Modes are:

- grey
- rgb555
- rgb24
- rgb32
- yuv422
- yuv411
- yuv420
- yuv410

Not all devices support all these modes.

```

string Device = ""
string Port = ""
int Width = 320
int Height = 240
string Mode = "yuv420"
    
```

vfg_video_in_fg	
DeviceFilter	out-{}-in
PortFilter	
Width	
Height	
Mode	

```

vfg_Display_fg
{}-in widget
    
```

VFG can be downloaded from

<http://gedae.com/downloads/releases.html>

Follow the installation instructions in the VFG\_installation.txt file in the downloaded folder.

## Gedae Table Enhancements

The Gedae Table capability has been enhanced in two ways. A search capability has been added to allow the user to quickly locate table entries by name. Also the text fields have been modified to allow users to type text directly into modifiable text fields. The enhancements to the gedae tables affect the following tables:

Partition Table

Auto Subschedule Table

Map Partition Table

Transfer Table

Firing Table

Memory Map Table

Queue Table

Schedule Info Table

Trace Table

Parameter Table

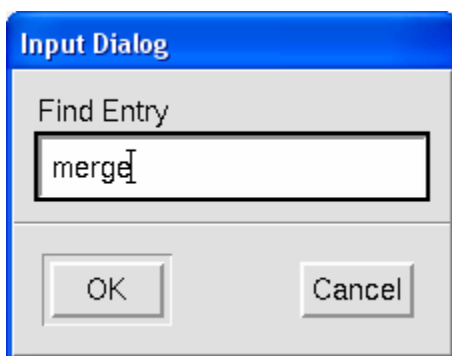
Hierarchy Table

Clock Table

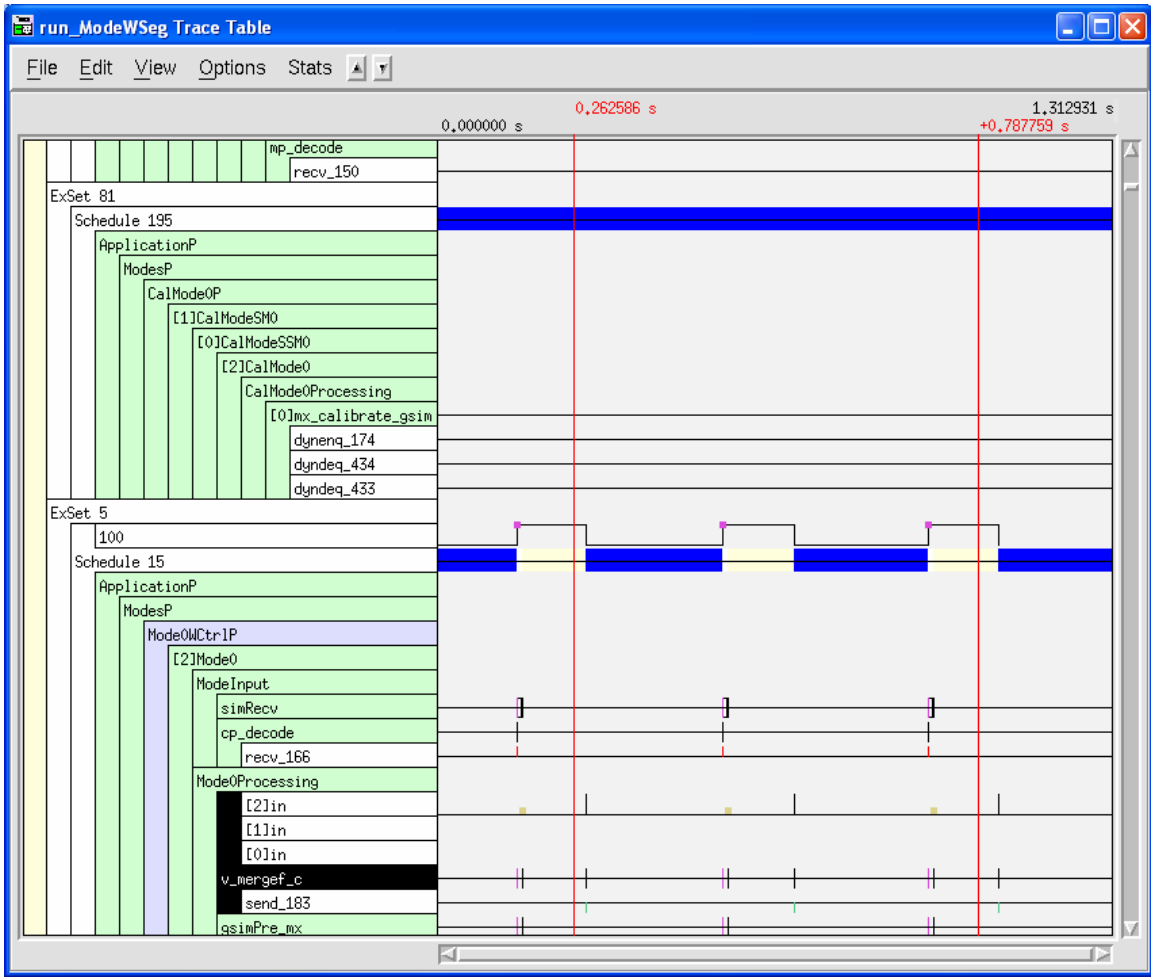
Delay Table

## Table Search Capability

The search capability allows the user to type in any part of a hierarchical primitive name. For example typing in:



Will find the first entry in the graph containing the string merge (as seen below).



Searches begin from the top of the table or from the currently selected table item. The box above would also be found if the user typed in any part of the name

ApplicationP.ModeP.Mode0WCtrlP.[2]Mode0.Mode0Processing.v\_mergef\_c.

that included the v\_mergef\_c. For example entering:

Mode0Processing.v\_mergef\_c

Will also match the above v\_mergef\_c box but won't match such a box if its in a subgraph named ModelProcessing.

## Direct Text Entry

Text can be directly entered in Gedae Table text fields. Below is the Mapping Table showing equation text being entered into the ProcNum column of Partition p1.

Name	CP	ProcNum	System Name	Trace Size	Trace MemType	Params
cp		100 *	gsim_host	1000000 *	default	
h		99 *	gsim_host	1000000 *	default	-d 0,001 *
p0		101=101+\$1	gsim_host	1000000 *	default	
p1		101=101+\$1	gsim_host	1000000 *	default	
p2		103=101+\$1	gsim_host	1000000 *	default	
p3		104=101+\$1	gsim_host	1000000 *	default	
p4		105=101+\$1	gsim_host	1000000 *	default	
p5		106=101+\$1	gsim_host	1000000 *	default	
p6		107=101+\$1	gsim_host	1000000 *	default	
p7		108=101+\$1	gsim_host	1000000 *	default	

Text can be entered either simply or “by equation”. The ProcNum cp entry shows a simple text entry of 100. The ProcNum p0 entry shows text entered by equation. In this case the entry applies to all of the partitions p0 through p7 that match the template p\$1. We will not reiterate the meaning of such equations here. The new capability provided with Gedae 5.0 is that such equations can be entered by typing directly into the text field. Text beginning with an “=” sign will be interpreted as an equation.

Toggle fields, such as the column CP below, can be toggled simply by clicking on the field.

Name	CP	ProcNum	System Name	Trace Size	Trace MemType	Params
cp		100 *	gsim_host	1000000 *	default	
h		99 *	gsim_host	1000000 *	default	-d 0,001 *
p0		101=101+\$1	gsim_host	1000000 *	default	
p1		102=101+\$1	gsim_host	1000000 *	default	
p2		103=101+\$1	gsim_host	1000000 *	default	
p3		104=101+\$1	gsim_host	1000000 *	default	
p4		105=101+\$1	gsim_host	1000000 *	default	
p5		106=101+\$1	gsim_host	1000000 *	default	
p6		107=101+\$1	gsim_host	1000000 *	default	
p7		108=101+\$1	gsim_host	1000000 *	default	

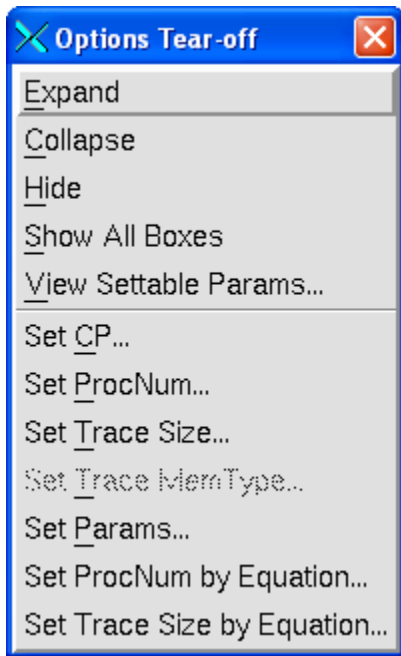
Here is a list of the meaning of different colors and symbols in the text fields.

- Fields with a white background are user settable.
- Fields with a gray background are not settable by the user but are for information only.
- Fields that are followed by an \* have been set by the user.
- Fields of the form <value>=<equation> (like the p1 ProcNum field 102 = 101+\$1) have been set by the user “by equation” where the equation appears on the right and the value the equation evaluates to appears on the left.
- Fields with black text that are neither followed by an \* or are set by an equation are set to their default values.

- Fields with red text have values that are derived from some other fields that have been set by the user.

Note that red text in with a white background may still be set by the user. That is the user can override derived values if the background color of the text says the text is user settable.

Any field that can be set by the user directly and/or can be set by equation will also appear in the Options menu. For example the Options menu of the mapping table is:



This menu shows that the CP, ProcNum, TraceSize and Params fields can be set directly and the ProcNum and TraceSize can be set by equation. These fields can be set both by using the Options menu and by direct entry into the text fields.

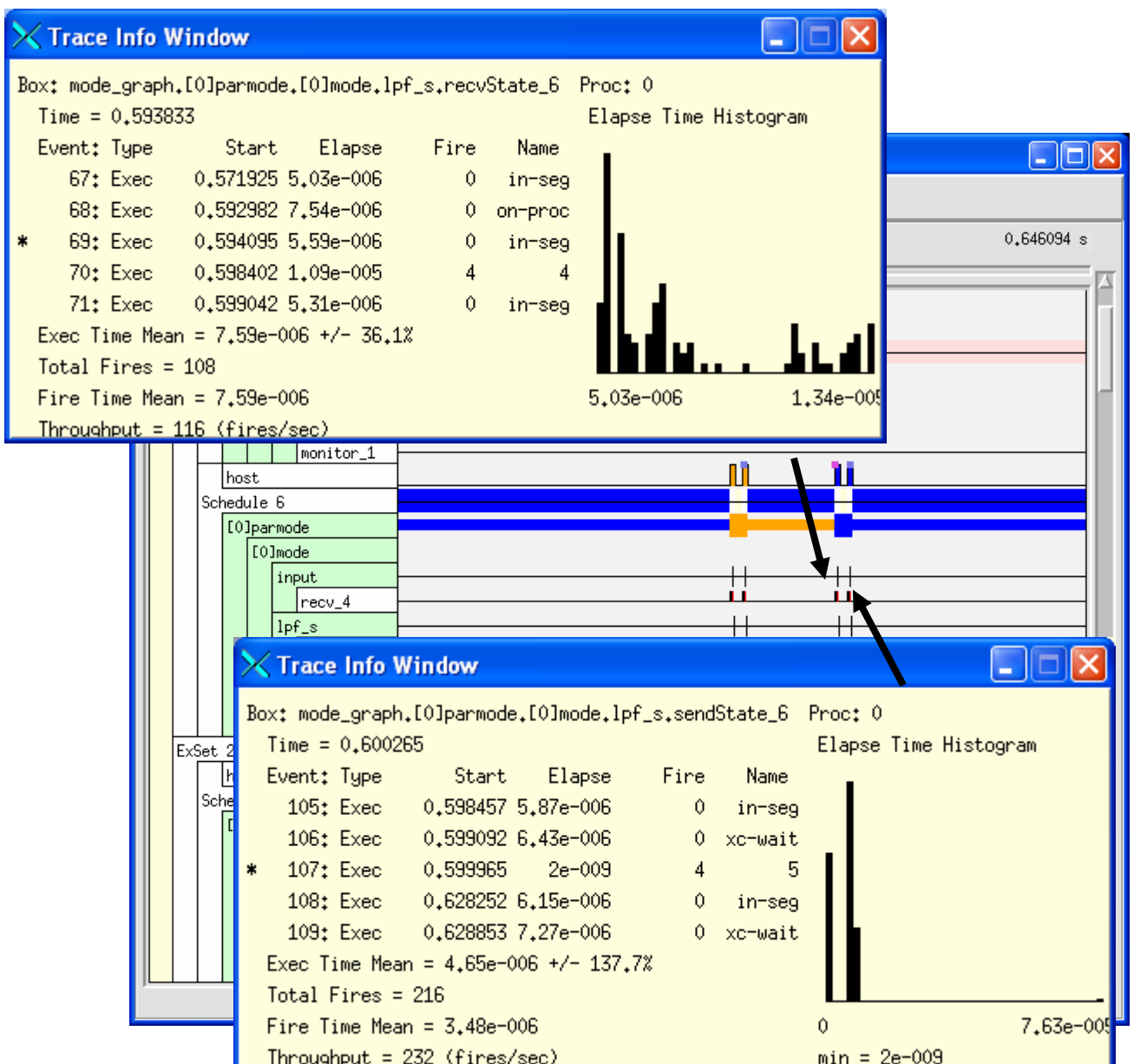
The text fields can be navigated either by clicking on the field and by using the arrow keys. All text entries into a field take affect as soon as the focus is removed from the text field.

### **SCR2193 Allow setting of parameters before appCreateFromLaunch completes**

Calls to appCreateFromLaunch can now be called in two parts - appCreateFromLaunchBegin and appCreateFromLaunchEnd. The appCreateFromLaunchBegin does everything except the parameter evaluation. This gives the user a chance to modify parameter values before appCreateFromLaunchEnd is called.

## SCR2367 Add ability to trace sendState/recvState Destination/Source

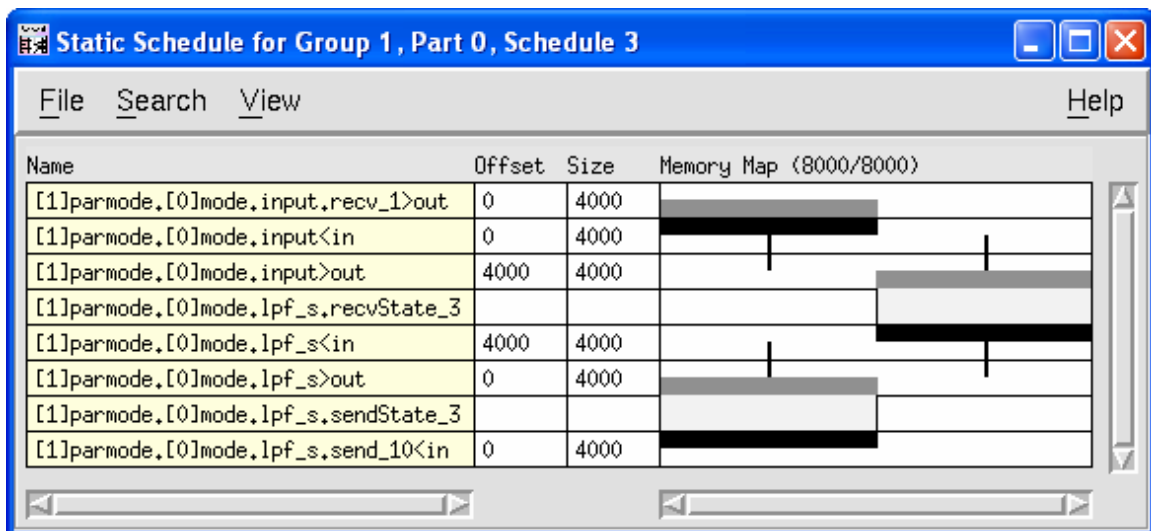
Graphs with external state variables that are distributed to multiple exclusive branches of a graph cause `sendState` and `recvState` primitives to be added to the schedules in which they are used. The `sendState` primitive will send the state to the next processor to use it and the `recvState` primitive will receive the state from the sending processor. The `sendState` primitives have multiple destinations and `recvState` primitives have multiple sources and every firing of these primitives may write/read from a different destination/source. The information about these changing destinations/sources is now recorded and can be retrieved from the trace table. The figure below shows the results of clicking the right mouse button on both a `sendState` and a `recvState` primitive.



The new information is given in the column labeled Name. For a `recvState` primitive a number in this column indicates the number of the `sendState` primitive that is sending data to the `recvState`. For example above the `recvState_6` primitive is receiving data from `sendState_4` at Event number 70. If the entry in the name column is `in-seg` this means that the `recvState` was called in the middle of processing a segment. Since state is only transferred on segment boundaries the `in-seg` designation means that the `recvState` is a noop. If the entry in the name column is `on-proc` this means that the `recvState` is being called at the beginning of a segment but the previous segment ran on the same processor as the `recvState`. Again this means that no actual transfer of data has occurred and the `recvState` is a noop. The `sendState` has the same possibilities. A number indicates the `recvState` that the `sendState` is sending to. Values of `in-seg` and `on-proc` mean that the `sendState` is a noop. If the entry in the name column of a `sendState` is `xc-wait` this means that the `sendState` has not yet received a command to transfer data to the next processor. In this event the `sendState` does not perform the transfer and acts as a noop. When the transfer command eventually arrives the `sendState` then sends the data to the `recvState` box. This sequence can be seen on events 106 and 107 above.

## SCR2242 Print Schedule Description

A new capability has been added to print a static schedule description in addition to the current capability to display the schedule description. The printed description is easier for users to send to the Gedae support team. To print a schedule, bring up the schedule display and the select `View->Print` to Terminal. For example the following schedule:



Name	Offset	Size	Memory Map (8000/8000)
[1]parmode,[0]mode,input,recv_1>out	0	4000	[Memory Map]
[1]parmode,[0]mode,input<in	0	4000	[Memory Map]
[1]parmode,[0]mode,input>out	4000	4000	[Memory Map]
[1]parmode,[0]mode,lpf_s,recvState_3			[Memory Map]
[1]parmode,[0]mode,lpf_s<in	4000	4000	[Memory Map]
[1]parmode,[0]mode,lpf_s>out	0	4000	[Memory Map]
[1]parmode,[0]mode,lpf_s,sendState_3			[Memory Map]
[1]parmode,[0]mode,lpf_s,send_10<in	0	4000	[Memory Map]

Gets printed as:

```

.-- SCHEDULE DESCRIPTION --. Group 1, Part 0, Schedule 3
|                               |
| Seg      Offset      Bytes | Name
|                               |
|-----|-----|-----|
| 0         0         4000 | [1]parmode.[0]mode.input<in
|                               | >out
|-----|-----|-----|
| 0         0         4000 | [1]parmode.[0]mode.input
|                               | <in
| 0        4000        4000 | >out
|-----|-----|-----|
|                               | [1]parmode.[0]mode.lpf_s<fb
|-----|-----|-----|
|                               | [1]parmode.[0]mode.lpf_s
| 0        4000        4000 | <in
| 0         0         4000 | >out
|-----|-----|-----|
|                               | [1]parmode.[0]mode.lpf_s<fb
|-----|-----|-----|
|                               | [1]parmode.mergef_c<[0]in
| 0         0         4000 | <in
|-----|-----|-----|

```

## SCR2331 Get configuration control status of all boxes in graph

The user can now get the configuration control status of all boxes (primitives and subgraph) in a top-level graph. To get the status, click on the File->Status All menu item. A dialog will popup reporting the status of all boxes in the graph.

## SCR2351 Allow user to change the command used to rsh to a remote Unix processor

A new processor argument `-rsh <rsh_command>` has been added to the esolaris and eredhat BSPs. This command allows the user to set the command that will start processes on the remote processor. For example, `-rsh ssh` will set the BSP to use the `ssh` command in place of the default `rsh` command. The parameter should be set in the embedded configuration file `Processor_Types` section `Info` field as:

```

Processor_Types: { redhat: {
  Type: "eredhat"
  Make_Params: "eredhat/runtime_make_info"
  Info: "-rsh ssh -ftp -user shlabotnik"
  Memory_Desc: {}
}

```

In the above example the `Info` field substring string `"-rsh ssh"` sets the command used to start a process on a remote processor to `ssh`.

## 2 Bugs Fixed

### **SCR2025 Undefined Initialization Function Symbols**

The problem occurs when the directory serial number table that is used to code generate primitives and to code generate the target executable is not in sync so the target executable refers to undefined symbols. An example is that the compile complains about an undefined symbol `I0002_add`. The code generated `add.c` primitive provides the symbol `I0004_add`.

### **SCR2143 Create dynamic dependency through nondet inputs**

A static schedule that exits to another and then reenters the static schedule may require a dynamic dependency to be created between the box at the exiting point and the box at the entering point. Without this dependency the box at the entry point may fire first which would be wrong. Prior to this release if there were nondet queues in the path from the source to the destination no dependency would be created because it would be possible for the boxes with nondet inputs to fire without the upstream boxes firing. Creating a dependency in this situation could cause a graph to deadlock. Now more analysis of the graph is done and sometimes these dependencies are created if it is guaranteed that a deadlock won't be created by them. Adding the dependency makes some graphs run that were not running before.

### **SCR2184 Compile can fail if dynamic variable vector dimensions not referenced in Apply method**

The code generator does not add declarations in the `Apply` method of variable dimension variables of a variable vector if they are not referenced in the `Apply` method. This is especially critical when the `amount` function is called as these references are needed. A related problem is that family indices that are not referenced in the `Apply` method are not declared. This causes an error for dynamic inputs and outputs.

### **SCR2186 Small hierarchical embedded configuration file causes segfault during attach.**

An embedded configuration file with few processor entries but with one processor entry that hierarchically pulls in another embedded configuration file causes the development environment to segfault while trying to attach the graph. The problem has been solved by assigning all primitives to run on the "host" by default rather than to give them a unique processor id.

### **SCR2190 Better Error Reporting for Dynamic Dependency Algorithm**

A case can arise where there are multiple nondet input boxes that might be able to fire. Gedae attempts to fire all of them and this may result in missing some dependencies. This situation should be reported.

## **SCR2191 Initial trace events lost when running attached command program**

When running the auto-generated exec-host command program from the development environment initial trace table events are lost. Events are lost because the command program starts running before the development environment has sent the command to start trace collection.

## **SCR2194 Use one symbol for data families in C-LSP for Gedae-RTL**

Families are expanded in the netlist so that a family of size 100 creates 100 symbols in the C file. With some common algorithms, this creates so many symbols such that the Windows compiler cannot handle them with debugging on. Give option to keep family of variable collected into 1 symbol with a dimension.

## **SCR2197 Dynamic Pointer Streams cause graphs to share data incorrectly**

The static schedule memory of a box that copies that memory to an output dynamic pointer stream may be corrupted.

## **SCR2198 embWallclock doesn't work for VxWorks Command Programs**

For VxWorks command programs written by a user the `embWallclock` command does not work.

## **SCR2201 Fix Save and Restore of Trace Table**

If a Trace table is saved (using the Trace Table menu item `File->Save Table`) and then restored (using the Gedae top-level menu item `Application->Open Trace Table`) Gedae segfaults during the restore.

## **SCR2204 e\_dv\_\_\_\_.h files prototypes name functions e\_v\_\_\_\_**

Many of the `.h` files used by primitive writers to include the prototype for vector double functions contain names like `e_vadd` instead of `e_dvadd`. As a result the functions actually used are not prototyped and this can lead to errors in passing arguments.

## **SCR2207 Launch package parameter initialization error**

A launch package that has derived parameter depending on only settable parameters will fail to calculate the derived parameter values. This problem has been fixed.

## **SCR2209 Speed up search utility startup**

Changes to the Search Utility in 4.9 caused startup of the dialog to be noticeably slower. To speed up launching of the dialog, the functionality descriptions are formed during the database generation instead of at the time of the first search. Because of this change, users who have generated their own search databases should regenerate the files.

## **SCR2212 Queue Resizing doesn't work for nondet inputs of segmented graphs**

The automatic runtime queue resizing algorithm fails if a nondet queue needs tokens and the schedule has been set to move to the end-of-segment state on completion.

## **SCR2213 Segfault when resizing queue whose memory space is shared**

If the runtime-resize-queues flag is set and a queue is resized whose memory is shared then Gedae will crash.

## **SCR2217 Improve Dy4AV2 BSP Wallclock Timer**

The `dy4av2 embWallclock` function used to collect trace table times has a bug. If it is not called for a long period of time (seconds) then the wrapping of the timer register may not be recorded. This causes a gap in the time record and may cause the trace table calibration to be incorrect.

## **SCR2218 Extend BSP Kit to include sleep/wakeup**

The BSP Kit does not define the functions `embGoToSleep`, `embWakeup`, `embSelf` and `embSetHostPortSchedule` and `embUncompress`

## **SCR2219 Flattened graph doesn't size correctly**

When the zoom factor on a flattened graph is changed, then flattened graph canvas is sometimes too small to see the entire graph.

## **SCR2235 Remove editor-raising buttons from top-level editor**

The raise top-level and raise parent editor buttons are not needed in the top-level editor. Remove these buttons.

## **SCR2238 Target Command Program with Host Partition doesn't byte swap correctly**

If a BSP requires byte swapping and a target command program partition is mapped to "host" then the BSP does not perform the byte swapping for the host partition causing the application to start incorrectly.

## **SCR2240 Gedae hangs after Run Kill Run Kill Run. of a distributed graph**

If a user does a `Control->Run` followed by a `Control->Kill` and then repeats the process and then does one more `Control->Run` on a distributed graph Gedae can hang (or possibly crash).

## **SCR2241 Primitive indices appearing as both families and dimensions are repeated in state vector**

If an index appears as both a family and a dimension as:

```
stream int [F]in;  
int a[F];
```

then it appears twice in the code generated primitives state vector causing the primitive not to compile.

### **SCR2259 X-Server locks up when View Data on unset parameter from Edit Popup menu**

Right click on an unset parameter to get the Edit Popup menu. Select `View Data` to bring up the Set Value dialog for the parameter. The Set Value dialog does not respond to mouse clicks. Successive invocations of `View Data` in this manner can lead to a segfault.

### **SCR2264 Reduce number of writes during sending of schedule**

When sending a schedule from a command program to a target processor the data is sent using many small writes. The number of writes is a linear function of both the number of memory blocks and the number of port names on the target processor. The number of writes has been reduced from the function:  $6 + 12*N_b + 4*N_p$  to  $6 + 4*N_b$  where  $N_b$  is the number of blocks (usually small)  $N_p$  is the number of ports.

### **SCR2299 Better I/O type checking in SFG's Edit/Add Data**

SFG allows "output out" to be specified, but it is not valid. Outputs must be streams. Inputs and outputs both must be typed.

### **SCR2309 mz\_inv gives incorrect results**

The primitive `embeddable/matrix/splitx/mz_inv` gives incorrect results.

### **SCR2310 Error in starting application with -loadlaunch when internal box used directly in graph**

When loading a previously created launch package that contained a primitive from the internal library - the development environment segfaults during the load.

### **SCR2313 Restore Trace Table segfaults if graph has external state**

If the trace table of a graph having external state primitives is saved then Gedae segfaults when the save trace table is opened.

## **SCR2319 If embedded configuration doesn't exist gedae segfault**

If there is not embedded configuration file appropriate for the processor Gedae is running on then Gedae segfaults. Also embedded configuration expects there to be a processor 0.

## **SCR2321 Avoid lengthy group state transitions needed to refresh tables**

When a table requiring a group to be in a certain state is displayed and a change to a group setting is made that requires the group to move to a lower state the group will move to the lower state and then back to the higher state in order to refresh the table. For example if the Schedule Info Dialog is displayed and a partition is remapped to a different processor the group is moved down to down to the APP\_CREATED state and then back to the SCHEDULED state at which point the Schedule Info Table can be refreshed. For large graphs this can take a long time. The solution is to measure the time that each group state transition takes. Then using this recorded information before a refresh of a table is requested the time required to move to the new group state is estimated. If the time is too long the table is closed and the group state transition is avoided.

## **SCR2324 Modify BSP to allow incremental archiving.**

If a graph has many different primitive types the launch package creation can fail. Failure occurs when creating the library that contains all of the primitives. Add ability to do incremental archiving of libraries.

## **SCR2325 Make read-only elements of FGTables selectable**

Some FGTable data does not fit in the column. If the column is read-only, you cannot read the cut-off data. If the data was selectable you could scroll the data horizontally with the mouse. Also, selected data can be cut and pasted.

## **SCR2326 Change asserts to traceable events followed by command handling in RTK**

Assert errors in the RTK caused the target process to abort. This behavior has been changed to instead:

1. Add trace events indicating that the problem has occurred. Trace events can be either added to input queues, output queues or schedules.
2. Signal the command program that a problem has occurred. Signaling the development environment causes a dialog to popup displaying the error.
3. The target then goes into a command handling loop. This allows the developer to view the trace table or other debug information if desired.

## **SCR2336 Distributed exclusive segmentation assert error (MAX\_LCS)**

If a graph has an exclusive distribution tree and the distribution is done at a deeper segmentation level than the collection on the other side then an assertion error can occur as:

```
assertSchedules(exl->nucs <= MAX_LCS,exls->ns,exls->s,  
                "Lock Count Exceeded");
```

This error is now detected as:

```
assertSchedules(exls->s[0]->nseg_ids>=oci.nseg_ids,exls->ns,  
                exls->s,  
                "Exclusive distributor is at a deeper\n"  
                "segmentation level than the collector");
```

## **SCR2340 vx\_MXmult and vz\_MXmult give incorrect results**

The primitives `vx_MXmult` and `vz_MXmult` give incorrect results. The call to `e_cmmul` and `e_zmmult` used the vector pointers in an incorrect order.

## **SCR2341 Group Settings Lost**

When a graph is modified the group settings for the graph will often not restore correctly. This problem has been fixed.

## **SCR2343 vi\_and incorrect on Solaris**

The primitive `vi_and` gives correct results on NT but incorrect results on Solaris. Solaris does not handle the pointer increments (using `++`) correctly.

## **SCR2346 Fix second level compression for targets with different endians than the host**

If the target processor has a different endian than the host processor then the input data to the second level compression algorithm must be 4 byte swapped before the algorithm is called and the output data must be 4 byte swapped after the algorithm is completed.

## **SCR2352 Subgraph Parameter Table does not close**

Subgraph parameter tables did not close when `File->Close` or window `Close (X)` button is selected.

## **SCR2353 nondet destinations of exclusive outputs do not set lock correctly**

Exclusive queues with `dest_qp` of `NREQ_Q` do not set the `ex_lock` field correctly causing this field to go negative.

### **SCR2354 Exclusive Queue with unconnected output causes segfault**

Exclusive queues must have all of their outputs connected. A check has been added during the preprocess phase to report this as an error that must be fixed in the graph.

### **SCR2355 EndOfSegment processing fails to put End-of-segment marker on locked queue**

If a queue in an exclusive branch is locked it may be in the waiting-for-begin state even though a segment begin has been added to the queue. In such a case if the source schedule runs its end-of-segment method it will fail to add an end-of-segment marker on the output queue. The locked case is now detected and the problem is fixed.

### **SCR2356 Cyclic delay can cause erroneous detection of dynamic dependencies loop.**

A cyclic box that on requires some but not all of its inputs to fire the first time may cause the dynamic dependency algorithm to not successfully fire the subgroup containing the cyclic box. This failure to fire the box causes an erroneous detection of a dynamic dependency loop and this detection causes the graph to fail to schedule.

### **SCR2365 Segfault while reloading group settings of edited graph.**

If an arc is deleted and then re-added (or a box is deleted and re-added or a box is replaced) and group settings are afterwards reloaded Gedae can segfault.

### **SCR2366 change embResume to not require protection**

The `embResume` function that is used to wakeup paused send/recv boxes and user created I/O devices previously had a bug that made it possible to miss an `embResume` if multiple resumes were called at the same time. This problem has been fixed. The only assumptions are that the interrupt procedure or calling thread that calls `embResume` cannot be interrupted by the main Gedae processing thread. The main Gedae processing thread however can be interrupted at anytime by the `embResume` thread. It is up to the BSP developers and anyone developing an I/O device to insure that the assumption is correct.

### **SCR2368 Detect exclusive primitive outputs that are not also families**

Exclusive outputs must also be family outputs. This is now enforced by the Gedae primitive parser. If the condition exists the primitive will not load and the following error is printed on the terminal: Error: Exclusive output out must be a family

### **SCR2369 Dy4 needs to sleep for a few seconds after Reset**

The Dy4av2 needs to sleep a few seconds after the Reset to allow the target-to-target ports that are created to be used. The capability to register and call a function after the

reset is complete has been added to the Gedae RTK. The dy4av2 now registers such a post-reset function in its embInit function to force sleeping for a couple of seconds.

## 3 Known Bugs

### **Case 1024: Accommodate Different Exceed Versions in `initGEDAE`**

In order for Gedae to work with Exceed/XDK 9.0, `makeGEDAE` has to be modified. Currently users must view the FAQ for 7.1 changes [http://www.gedae.com/SUPPORT/FAQ/exceed\\_7\\_1.html](http://www.gedae.com/SUPPORT/FAQ/exceed_7_1.html) and then remove `xlibcon` from `makeGEDAE` and `ent` files mentioned in the FAQ.

### **Case 1026: Broadcast Transfer Mechanisms**

Currently all Gedae data communication is point to point; however, many target processors support efficient broadcast mechanisms. The goal of this task is to extend the Gedae BSP API to allow broadcast data transfers to be part of the BSP. A second goal is to fix the DSA mechanisms that cannot be implemented when there is fan-out (see Case 1149). Rather than fixing that problem directly, a broadcast DSA capability will be implemented.

### **Case 1056: 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.

### **Case 1057: 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.

### **Case 1078: Gedae Overwrites Protected Files**

Gedae overwrites files that are protected as a result of being checked out of a CMS. This overwriting causes a problem because it defeats the CMS system.

### **Case 1108: Setting Default Subschedule Queue Capacities Correctly**

The queue policy and capacity of a queue feeding a subschedule may not be set correctly.

### **Case 1122: 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.

### **Case 1139: Unterminated Comments**

Unterminated comments cause the Gedae parser to segfault.

### **Case 1140: Parser Problem**

The Gedae parser does not handle an odd number of quotes (") well.

### **Case 1142: Arrays of Strings Not Allowed**

Gedae currently allows string array graph parameters to be declared as:

```
const string X[] = {"hello", "world"}
```

or

```
string X[i] = [i]Y
```

where  $Y$  is a family of strings.

In either case, the values so declared are not correctly set, and therefore, should be considered illegal.

### **Case 1143: Inconsistent Data Type Declarations**

Gedae aborts when the same stream type is multiply defined. For example, suppose two primitives define data types with the same name but different definitions. If the primitives use these types in their `Input`, `Local` or `Output` sections, then Gedae aborts.

### **Case 1144: Function `appFree` Memory Leak**

A command program running on VxWorks does not free all the resources allocated (memory, sockets, etc). The `appFree` function must release everything allocated. Gedae should automatically generate a call to `appFree` for the standard `exec-host` command program.

### **Case 1145: External Code Does Not Recompile**

`Make` is not called after a successful run, so changes to code listed in the `Personal_Emb_Obj_List` do not get recompiled. To force the recompile, it is currently necessary to change something from the Gedae GUI – such as, saving a primitive or toggling the Group "Run on Embedded" toggle off and on.

### **Case 1146: Large Graphs Fail to Display on Flattened Graph**

If a graph is too large, then it cannot be displayed on the flattened graph. That is, if the flattened width or height exceeds the allowable pixmap width or height.

### **Case 1147: 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.

### **Case 1148: VxWorks `embWallclock` Function Misses Wrap**

When collecting trace information, the `embWallclock` function timer can wrap without being detected. This failure to detect the timer wrap causes VxWorks processor timelines to appear compressed.

### **Case 1149: 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.

### **Case 1150: Trace Table Saved on NT Not Readable on Solaris**

Files saved from NT are byte reversed from what is expected on Solaris. Files saved on a big-endian platform cannot be read by Gedae from a little-endian platform.

### **Case 1151: 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.

### **Case 1155: Constants Propagated Through typedef Boxes**

Constants propagated through typedef boxes cannot be used for instantiation.

### **Case 1158: Primitives with `EndOfSegment` and No `Apply`**

Primitives that have an `EndOfSegment` method but do not have an `Apply` method do not get included; therefore, the `EndOfSegment` method does not run. A workaround for this problem is to include an empty `Apply` method in the primitive.

### **Case 1159: 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.

## **Case 1239: Static Schedule in Segmented Subgraph Not in Scope of Segment Controller Asserts**

An assert error is caused by a static schedule in a segmented subgraph that has no input in the scope of the segment controller. For example, a constant box driving a merge input in the subgraph can cause the error. The error manifests itself as an assert as shown below:

```
Error: assert(dq->eos[inptr].begin_level > 0) failed.  
File: ../../source/segment.c, Line: 935
```

Such static schedules should be declared to be illegal at scheduling time.

## **Case 1249: 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.

## **SCR2012: 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.

## **SCR2015: 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.

## **SCR2019: Graphs with Host to Target Control Ports Fail on Linux and Solaris**

The problem is that the host is not performing mail box services while it is waiting to establish a control port to the target processors. Unix processors require the host to be performing these services in order to make connections.

## **SCR2022: 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.

## **SCR2024: 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.

## **SCR2025: Undefined Initialization Function Symbols**

Occasionally when linking a target executable, the linker complains about undefined function names such as `I0002_add`. A workaround for this problem is to do a `makGEDAE CLEAN` followed by a `makeGEDAE`.

## **SCR2047: 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.

## **SCR2054: 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.

## **SCR2062: 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.

## **SCR2064: 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.

## **SCR2065: 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.

## **SCR2078: 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.

## **SCR2104 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.

## **SCR2113 Failed Launch Package Creation**

The "failure making launch package" error is received when an absolute path is entered in the directory field of the Launch Info Dialog.

## **SCR2116 Segfault when "Map Parts" Fields Are Navigated While the Graph Is Running**

Gedae segfaults if the Map Parts Dialog fields are navigated while the graph is running.

## **SCR2131 Improve Flattened Graph Placement**

| After expanding the flattened graph, reduce the gaps between the boxes. There is currently a great deal of extra white space in the expanded version of the flattened graph.

## **SCR2157: Need to Turn on Launch Package Tracing**

Launch packages by default start with tracing turned off. Attaching the development environment to the running application may turn the tracing on but any events before attaching are missed.

### **SCR2158: Trace Table send/recv Webs Do Not Work from an 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.

## 4 BSP Kit Changes

### Change to `std_make_info` `EMB_ARCHIVE` variable

The `$GEDAE/<host>/<target>/std_make_info` file `EMB_ARCHIVE` environment may need to be changed. When creating an archive the `EMB_ARCHIVE` variable is now immediately followed by the name of the output file.

For example if the `EMB_ARCHIVE` command is the `gnu` archiver then the `EMB_ARCHIVE` was previously set as

```
$ENV{"EMB_ARCHIVE"} = "ar rc";
```

This must be changed to:

```
$ENV{"EMB_ARCHIVE"} = "ar rc ";
```

On windows platforms the `ent` BSP archiver is defined as:

```
$ENV{"EMB_ARCHIVE"} = "lib -out:";
```

This remains unchanged as the `lib` command does not expect a space between the `-out:` and the name of the output library.

### New `std_make_info` `EMB_INCREMENTAL` variable

A new environment variable `EMB_INCREMENTAL` may be optionally defined to indicate that the archiver specified by `EMB_ARCHIVE` can be used for incremental archiving. The `EMB_INCREMENTAL` variable can have a value of 0 or 1. If the value is 0 then the incremental archive command is formed as:

```
$EMB_ARCHIVE<libname> <objectfile>
```

For example

If the value is 1 then the incremental archive command is formed as:

```
$EMB_ARCHIVE<libname> <objectfile> <libname>
```

If `EMB_INCREMENTAL` is not defined then all libraries are created by providing a complete list of object files to the archive command.

## **Change to link\_makefile**

The rule to create `LIB_NAME` is no longer used in the `link_makefile` and may be removed.

Add `$(TARGETNAME)_up_to_date` to `EXEC_NAME` and `COMM_NAME` dependency list