

Design Creation Release Notes

Mentor Graphics Release B.1



Copyright © 1991 - 1995 Mentor Graphics Corporation. All rights reserved.
Confidential. May be photocopied by licensed customers of
Mentor Graphics for internal business purposes only.

The software programs described in this document are confidential and proprietary products of Mentor Graphics Corporation (Mentor Graphics) or its licensors. No part of this document may be photocopied, reproduced or translated, or transferred, disclosed or otherwise provided to third parties, without the prior written consent of Mentor Graphics.

The document is for informational and instructional purposes. Mentor Graphics reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Mentor Graphics to determine whether any changes have been made.

The terms and conditions governing the sale and licensing of Mentor Graphics products are set forth in the written contracts between Mentor Graphics and its customers. No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Mentor Graphics whatsoever.

MENTOR GRAPHICS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

MENTOR GRAPHICS SHALL NOT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS PUBLICATION OR THE INFORMATION CONTAINED IN IT, EVEN IF MENTOR GRAPHICS CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

RESTRICTED RIGHTS LEGEND Use, duplication, or disclosure by the Government is subject to restrictions as set forth in the subdivision (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

A complete list of trademark names appears in a separate "[Trademark Information](#)" document.

Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070.

This is an unpublished work of Mentor Graphics Corporation.

TABLE OF CONTENTS

About These Release Notes.....vii

Chapter 1

Design Architect

V8.5_1.8.....1-1

Known Problems and Workarounds.....1-1

Corrected Problems.....1-2

Defects Corrected in This Release.....1-2

Changes.....1-12

New Functions.....1-12

Changed Functions.....1-12

“No Connect” Symbol Ability.....1-12

Enhancements.....1-13

DA enhancements for the PCB process.....1-14

New Editing Functionality.....1-15

Chapter 2

Design Viewpoint Editor

V8.5_1.8.....2-1

Known Problems and Workarounds.....2-1

Corrected Problems.....2-1

Defects Corrected in This Release.....2-1

Changes.....2-3

Changed Functions.....2-4

New Function.....2-5

Enhancements.....2-5

Chapter 3

Conserve

V8.5_1.8.....3-1

Known Problems and Workarounds.....3-1

Corrected Problems.....3-2

Defects Corrected in This Release.....3-2

Changes.....3-3

TABLE OF CONTENTS [continued]

Enhancements_____	3-3
Chapter 4	
Design File Interface	
V8.5_1.8_____	4-1
Corrected Problems_____	4-1
Patches Included in This Release_____	4-1
Defects Corrected in This Release_____	4-1
Enhancements_____	4-2
New DFI Functions_____	4-2
Chapter 5	
QuickCheck	
V8.5_1.8_____	5-1
Known Problems_____	5-1
Corrected Problems_____	5-1
Defects Corrected in This Release_____	5-1
Changes and Enhancements_____	5-2
Chapter 6	
CDP	
V8.5_1_____	6-1
Known Problems and Workarounds_____	6-1
Corrected Problems_____	6-1
Defects Corrected in This Release_____	6-1
Chapter 7	
EDIF (ENRead, ENWrite)	
V8.5_1.1_____	7-1
Known Problems and Workarounds_____	7-1
ENRead_____	7-1
ENWrite_____	7-1
Corrected Problems_____	7-2

TABLE OF CONTENTS [continued]

Patches Included in This Release_____	7-2
Defects Corrected in This Release_____	7-2
ENRead_____	7-2
ENWrite_____	7-2
Changes_____	7-4
Enhancements_____	7-4

TABLE OF CONTENTS [continued]

About These Release Notes

This document contains release notes for the Design Creation applications.

Items in “Known Problems and Workarounds” and “Corrected Problems” subsections for each application refer to a PR (Product Report) number or a DR (Defect Report) number. Some items may contain both numbers. The numbers simply reference two different MGC-internal tracking systems, the older Product Report system and the newer Defect Report system. Bugs that were reported and fixed before the Defect Report system was put in use have only a PR number. Bugs reported before the Defect Report system was put in use but fixed after the new system was in use contain both PR and DR numbers. Bugs reported after the Defect Report system was put in use have only a DR number.

You may be using other Mentor Graphics products not covered in this document. Consult the release notes for those specific products. Other release note documents that may be of interest are the *Falcon Framework Release Notes*.

If you are using this document on-line in INFORM, you will see occasional highlighted text. On a black and white display, this text appears enclosed in a rectangle, and on a color display using the default color map, the text is blue. The highlighted text is a hypertext link to related materials in this and other documents. If you click the Select mouse button on a hypertext link, the linked location will be displayed. For more information on hypertext links in on-line manuals, refer to the section “[Navigating Online Documents](#)” in the *BOLD Browser User's Manual*.



For information about the documentation conventions used in this manual, refer to *Mentor Graphics Corporation Documentation Conventions*.

Chapter 1

Design Architect V8.5_1.8

Known Problems and Workarounds

This section lists the known problems in Design Architect V8.5_1.8, and describes workarounds where available.

- **DR 113665**

DA allows two different NET props on the same net.

Description: The customer has found a situation where DA allowed two different NET properties with different values to coexist on a single net at the same time. It is not known how this occurred.

This problem has caused significant problems downstream (Package) due to bizarre net shorts that were difficult and time-consuming to track down.

Workaround: None known.

- **DR 111639**

Second open down loses lower level annotations

Description: There are times when after opening down on a sheet, closing the sheet, then opening down again that the annotations will not appear after the second open down.

Workaround: Close the current DA session and start another. You should not save any of the design sheets currently open in that session.

- **DR 108857**

Unlatch sub-tree doesn't unlatch design

Description: This problem occurs if DA is used to edit a symbol, then is left open while a viewpoint which uses that symbol has its latch updated. Then, if the symbol is again edited in the same session of DA, it will write the latch out pointing to the wrong version.

Workaround: Do not leave DA invoked when updating the latches of a viewpoint in DVE.

- **DR 96048**

DA crashes after doing symbol edit in design context view.

Description: If a sheet is opened in design context, and an edit symbol in place is done on one of the instances on the sheet a crash may occur. The crash will happen if the edited symbol has pins removed, and a save is attempted on the design sheet.

Workaround: Do not remove pins from symbols that are currently being referenced in an open design sheet.

Corrected Problems

Defects Corrected in This Release

The defects listed below have been corrected in this version of Design Architect.

- **DR 98993**

Plots are missing circles.

Description: The correct color map was not being used when printing. This resulted in the wrong colors being used on many devices.

Resolution: The correct color map is now being used.

- **DR 44723**

Copy needs option to propagate new component name to the symbol below

Description: When parts (symbols or components) are created within Design Architect, by default, the container name will be the same exact name as the symbol within the component. This default naming convention for component containers and their symbols occurs if you don't specifically request a different file name for the symbol.

Since this is a default behavior of Design Architect, the customer would like an option when changing names, moving or copying a component container, such that the new name of the container would propagate to the symbol (providing, of course, that the container name and symbol data name matched).

Resolution: The solution to this problem is to have a new set of post move, copy, and rename triggerware for `mgc_component` that renames the symbol and interface inside of a copied component to match the name of the new component. However, this renaming can only be done if the old symbol name is the same name as the old component.

The new `mgc_component` triggerware will rename the symbol and interface name to match the new component if there is exactly one symbol in the component, and that symbol has the same name as the component (in this case it is assumed that the interface name also matches the component name), or there are no symbols in the component, and there is exactly one schematic model.

If the new behavior is not desired, it can be disabled by setting the environment variable "MGC_NO_INTERFACE_RENAME" to any value.

- **DR 84570**

`$set_bgd_color` does not set the background color.

Description: When setting the background color for a symbol a crash occurs if the specified color is not in color map.

Resolution: Check to make sure the color is valid before trying to set it, and issue an error message if it is not valid.

- **DR 112065**

pinlist file does not accept ‘;’ in pin names

Description: The pinlist file used by DA to generate a symbol does not allow for semi-colons in pin names. Thus, you cannot generate a symbol with a multi-dimensional pin.

Resolution: ‘;’ is now a legal character to have in a pin name, and the symbol will be generated with the correct pin name.

- **DR 111040**

DA crash upon check sheet, owner after deleting net property

Description: When a net property which has been propagated from a pin is deleted, a check sheet can cause a crash.

Resolution: Added checks to avoid using NULL pointers.

- **DR 109491**

slash key on open sheet deletes the sheet attribute file

Description: If an existing sheet is opened, and an extra slash (“/”) character is specified so that the sheet name looks like “sheet1/”, then DA will issue the message:

Error: Cannot create mgc_sheet at path...

and deletes the sheet1.mgc_sheet.attr file. This makes the sheet unusable and design data is lost.

Resolution: If a user enters a trailing slash, that slash will now be ignored and the sheet will open as expected.

- **DR 99378**

SIGIOT if non-existent version of a sheet

Description: Trying to open an old version of a sheet (cable or DA) which does not exist, DA spits out bunches of errors stating that a signal trap has been encountered (IBM only).

Resolution: The problem has been fixed.

- **DR 76115**

Property ownership incorrect after successive instantiations

Description: After multiple instantiations of two symbols with different property owners for the same pin property, instances of both symbols end up with the same ownership for that property. Thus, the pin properties for instances of one of the symbols now has the wrong ownership.

Resolution: This DR has been fixed by having the part pin properties remember if the property is to be propagated to the net. The property name information is no longer used to determine net vs. pin placement.

- **DR 111061**

Check schematic errors with Signal 4, illegal instruction code 9

Description: This problem only happen when there was non-scalable text on a sheet, and that sheet was not open during a check schematic. During the check the property was being accessed, and a pointer to the sheet was used to do a calculation. However, since the sheet was not open it failed.

Resolution: Make sure the sheet the property lives on is open before trying to do the calculation.

- **DR 72987**

Check schematic on a read-only sheet hangs

Description: If a check schematic is run while one of the sheets of the schematic is open read-only, DA will hang.

Resolution: DA was fixed to not hang in this situation.

- **DR 107001**

Saved Data with errors cannot be opened as design sheet

Description: If a sheet is saved with errors (e.g. a ripper rule of “R” is left unset), then a design sheet can not be opened on that sheet.

Resolution: Designs sheets have been modified to allow sheets with errors to be opened.

- **DR 102021**

DA check sheet crashes - caught signal 10 - SIGBUS: bus error

Description: There were situations where deleting a NET property would put the data in an inconsistent state. Doing a check sheet would trip over that inconsistent state and cause a crash.

Resolution: The code which handles deletion of NET properties was modified to make sure the data was consistent.

- **DR 99214**

open design sheet crashes DA if the component interface changes

Description: Customer wanted to print the design sheets for a circuit they have in production. The design contained a part that had the pin order change. When they tried to do print design sheets, DA crashed.

Resolution: DA was not correctly setup to handle an error condition during print_design_sheets. DA has been modified to handle the error condition correctly and allow the operation to continue.

- **DR 85890**

Protected property is not protected from deletion

Description: A property that is protected on the symbol can be deleted by using the delete stroke, or hitting the “delete” key once it is selected. If the deletion is

attempted by using properties>modify and choosing the delete option the operation is not allowed.

Resolution: All places which attempt to delete a property now check to make sure it is not fixed or protected.

- **DR 66309**

Open down within context of design crashes DA

Description: When a Design Sheet is opened that has an instance whose pins do not match the interface, opening down on that instance will cause DA to crash.

Resolution: DA was not correctly handling the error conditions that resulted from the open down. It now handles the error conditions and allows the open down.

- **DR 115859**

BA visibility incorrect for instance-only source properties

Description: The P165 patch to fix BA property visibility does not correctly address instance-only source properties. If the source sheet has a instance-only property with the visibility set to hidden, a back-annotation to that property will show up as visible.

If the property originated on the symbol, and it is set to hidden, then the back-annotation visibility will be set correctly.

Resolution: The annotation attribute sections of DA have been re-written for B.1 to take care of this (and other) problem.

- **DR 72845**

Concurrent sheet editing: cannot save sheet

Description: When multiple users try to concurrently edit different sheets in the same schematic, it is possible for one of the users to be unable to save their sheet.

This problem occurs under the following circumstances:

- When one user opens a NEW sheet in the schematic, and another user saves a sheet in the same schematic BEFORE the first user save his sheet.
- The first user is able to use Save Sheet As to save the sheet to another location, so his work on that sheet will not be lost.

Resolution: DA now recognizes this situation and allows the new sheet to be saved.

- **DR 50873**

Update symbol problems with generate_symbol, make_symbol, save_symbol_as

Description: There was a problem with updating a symbol in the following scenario:

1. Open down on a symbol with one pin and create a new sheet below, which as a result, adds a number of ports.
2. Use the **Miscellaneous > Create Symbol** utility to build a new version of the symbol with the new pin names.
3. Check and save the symbol and then return to the schematic.
4. Select the symbol and do an update-auto. DA reports that the symbol is already at the latest version.
5. Use instance select to try a replace, and DA calls the one-pin version. The only way is to use **Edit > Begin Edit Symbol**, which grabs the new version.

Resolution: When an existing symbol was being replaced by `generate_symbol`, `make_symbol`, or `save_symbol_as` the version was not being rolled. The version is now rolled

- **DR 89453**

Extent of empty sheet is not correct

Description: The extent of an empty sheet, as reported by the `$get_sheet_extent()` function, is reported as being very large (see example below). Instead, it should report as being zero, i.e. `[[0,0],[0,0]]`

Resolution: The extent of an empty sheet is now returned as `[[0,0],[0,0]]`.

- **DR 82872**

vhdl plain_text files don't retain original protections when edited

Description: The vhdl plain_text file created/edited by the `$open_vhdl()` command or the `$$open_source_code()` command in DA does not retain the original UNIX file protections of the creator after being edited by someone else.

Resolution: VHDL plain text files now retain their original permissions when saved.

- **DR 65392**

More cntl of new BA prop text, beyond visibility

Description: Users would like to be able to control other TEXT parameters beyond just visibility - text size, orientation, justification, location, and rotation.

Resolution: A new command has been added to DA to give full control over annotated property attributes. The new function is `$setup_annotated_property_text()`. Refer to the DA reference manual for a full description of the function.

- **DR 27802**

Need stretching/resizing of graphic objects

Description: Currently, once a graphic object is created (particularly symbol/comment polygons) it cannot be resized, etc., but must be deleted and a

new object of the desired shape must be created. This is particularly a pain when you have properties on the object.

Resolution: It is now possible to stretch a symbol or comment object, and retain the properties associated with it. The new function \$stretch() has been implemented. A full description of the \$stretch() function can be found in the DA reference manual. The \$stretch() function can be accessed on the palette in the same location as the “scale” function used to be.

- **DR 46877**

Cannot open design sheet after check schematic

Description: After performing a Check Schematic on a design sheet, you are unable to open sheets under another design viewpoint even though you have closed all of the open design sheet windows.

Resolution: It is now possible change to a different viewpoint after doing a check schematic in design context.

- **DR 116452**

Check schematic - many warnings about unneeded offpage connectors

Description: When in design context and opened down from a top-level design sheet, Check Schematic produces many erroneous warnings about unneeded offpage connectors for many different nets in the design. Also, Check Schematic fails to detect instances across sheets with duplicate “INST” properties when executed in design context mode from a lower level of the heirarchy.

Resolution: The false check warnings are no longer produced, and duplicate “INST” properties are now detected.

- **DR 117115**

Open Down form not showing all sheets

Description: When opening down on an instance in design context from a sheet below the top level of the design, the Open Down form which appears does not list all of the available sheets.

Resolution: The Open Down form was using the list of models for the instance at the top level of the design with the same handle as the one being opened at the lower level. The form needed to append the design context of the instance being opened to get to the correct level in the hierarchy.

- **DR 117281**

Inst Property does not match Inst Handle after inter-diagram move.

Description: Inter-diagram move of an instance with an INST property of I\$0 incorrectly changes the INST property but maintains the same instance handle. Thus, the INST property and the instance's handle do not match.

Resolution: The resolution was to not change the INST property when the instance is moved.

- **DR 117625**

FalseError "Invalid property value syntax" in check sheet.

Description: If expression checking is turned on, and there are properties on a sheet which are either Triplet or Triplet Case type properties, then a false error message will be issued about the syntax of the property text not being correct.

Resolution: The syntax checking of Triplet and Triplet Case properties was updated.

Changes

There have been a number of new functions added, and existing functions changed in this release. Following is a list of the new and changed functions. A complete explanation of each function can be found in the *Design Architect Reference Manual*.

New Functions

\$set_check_annotations()	\$set_ripper_mode()
\$get_check_annotations()	\$get_ripper_mode()
\$reconnect_annotations()	\$set_ripper_query()
\$clear_unattached_annotations()	\$get_ripper_query()
\$setup_annotated_property_text()	\$setup_ripper()
\$apply_edits()	\$allow_resizable_instances()
\$get_apply_edits_needed()	\$get_allow_resizable_instances()
\$setup_default_viewpoint()	\$change_instance_resize_factor()
\$get_default_viewpoint()	\$get_instance_resize_factor()
\$get_bundle_members()	\$add_property_to_handle()
\$set_implicit_ripper();	\$stretch()
\$get_implicit_ripper()	\$slice()

Changed Functions

\$\$check()
 \$setup_check_sheet()
 \$set_viewpoint()
 \$set_autoripper()

“No Connect” Symbol Ability

In pre-B.1 DA, an instance with CLASS=N cannot have any pins. If a symbol is created with pins and a CLASS=N property, then check symbol will generate an Error, and the symbol can not be instantiated.

For B.1, the symbol editor has been changed to only issue a warning when an instance with a CLASS=N property has pins. This will allow the symbol to be saved and used on a sheet. Since the symbol has a CLASS=N property it will not be written to the electrical database, but it will stop dangle warnings in DA

Enhancements

Enhancements for Design Architect have been focused in two main areas: process improvements and editing functionality. The process improvements specifically target concurrent design in the board process and tie in to the board process white paper. The editing improvements deliver some of the most frequent customer requested enhancements.

- **Bundles** - The ability to group an arbitrary set of signals together by name.
- **Implicit Rippers** - This allows nets to be ripped from either bundles or buses by name rather than the ripper symbol and rule property. New userware gives a list of the bus or bundle members and then automatically places the name as a net property on the new net.
- **Stretchable Symbols** - In the symbol editor, just select a single object such as polyline or rectangle and stretch. A drag image is drawn of the stretching object. All properties remain intact and connected
- **Repeating Instances** - A short hand for frame for use with single instances.
- **Re-sizable Instances** - When enabled this function allows the user to shrink or grow an instance without modifying the symbol.
- **One Open Sheet** - There is now only one open sheet icon that recognizes whether a viewpoint has been set. The correct form for either open sheet or open design sheet is then presented.
- **Edit of SLD properties** - When DA is editing in design context, changes to SLD properties (net, inst, rule etc.) will automatically be made on the source sheet.
- **Check, Reconnect and Clear of unattached annotations** - Unattached annotations are no longer cleared from the back annotation files. New functions in DA allow them to be found then either re-attached or cleared.
- **Better protection for fixed and protected properties** - DA will not permit annotations to be made to fixed or protected properties. Downstream tools CAN make annotations to fixed properties and DA in Design Context will

warn in check sheet if this has happened. Merge annotations will generate a warning if an attempt is made to merge to a fixed or protected property.

DA enhancements for the PCB process

The enhancements described here have been specifically implemented to improve productivity in the board process; however, the techniques could be applied to any process that uses annotation files to reflect changes between team members. A full description of the board process is described in the board process white paper and is not detailed in this document. Suffice it to say that the B.1 enhancements allow both circuit design and layout to progress concurrently. Design consistency is maintained through the use of the synchronize process which reconciles changes between one or more designers and layout. The synchronize process is executed whenever it is required to bring the design to a consistent state. Simply put, the process allows the circuit engineers to see changes made by the layout designer while also taking any circuit changes forward into layout.

The following list of enhancements together provide the functionality required for the concurrent board process:

Reconcile: This is the utility which is used by synchronize to reconcile the Engineering Changes Annotation data into the PCB Changes Annotation. It is also the utility used to reconcile the Personal Changes annotations between multiple electrical engineers.

Allow edit of SLD properties during “Edit in Design Context”: When DA is editing a “design” sheet it isn’t possible to edit SLD properties (inst, net, etc.) while annotations are turned on. DA has been changed so that it automatically applies the edits in source mode to the sheet without the user having to manually change edit mode.

Retain annotations on copy or move: When an object is moved between “design” sheets, the annotations stay with the object. When an object is copied the annotations are copied with the object.

Protection of Fixed and protected properties: DA needs to respect the fixed and protected flag everywhere. Previous to B.1, annotations could be made to fixed or protected properties without warnings. At B.1 it will not be possible to

annotate a change to a fixed or protected property. DA checking will issue a warning when it finds a back annotation to a fixed or protected property. In addition, merge annotations will issue a warning when a merge has failed because of a fixed property.

One Open Sheet in DA: The presence of an open sheet and an open design sheet in the session palette has been a constant source of confusion for users, particularly those just using source sheets who don't know or care about viewpoints. This is now simplified to one open sheet icon. The open design sheet icon in the session palette has now been replaced with a set viewpoint icon. This sets the viewpoint and calls open sheet. The open sheet icon now recognizes whether a viewpoint has been set and opens the correct form for either open sheet or open design sheet. A button in the open design sheet form allows the user to switch back to open sheet.

New handling of unattached annotations: Annotations can become detached for a variety of reasons, for example an instance may be deleted and replaced. Previous to B.1, any annotation that have been applied to objects that no longer exist in the design are lost. This is a problem particularly if an annotation file is attached to two different viewpoints that have different primitive levels. If the design is saved from the design with the higher primitive level, the annotations at the lower level will be lost.

DA has new functionality to check for any “orphaned” annotations and then allow the user to either re-attach them to another instance or clear them from the annotation file.

Sheet based annotation locking: This allows multiple designers to edit different sheets of the same design in “design context” with the same viewpoint set. Previously, unwanted locking of the viewpoint would cause edits to be lost.

New Editing Functionality

Bundles: This is a new type of bus for DA that allows the grouping together of named nets. The object is referred to as a net bundle. The same kind of structure exists for pins and is called a pinbundle. The bundle may have a name followed by an ordered list of the contained signals: control{clk1,clk2,reset,s1,s2}. Cross sheet and inter-sheet connection can be made by the bundle name without having to re-

specify the contents: control{}. A defined netbundle applies to a complete schematic but is not in itself an electrical object which can span hierarchy. Pinbundles must be used for this purpose and care must be taken to maintain the ordering of the bundle contents which is used for the pin to net connectivity.

Implicit Rippers: Implicit rippers provide the ability to rip a named net from a bus or bundle by just connecting it to the bus. There is no need for the ripper symbol that is currently used. The name of the connected net must match one of the contained bits of the bus. Implicit rippers are intended to be one bit wide; for wide to wide connection, the current ripper symbol must still be used. A new “setup ripper” icon has been added to the add/route palette to make it easy to switch between ripper types.

New userware has been added for when an un-named wire is started or finished on a bus or bundle. A form will show the contents, allowing the user to choose which element they wish to rip. The net name is then automatically placed on the vertex next to the bus.

Re-sizable Instances: This allows the shrinking or growing of an instance by a factor of 2 or 4 without changing the symbol. This functionality needs to be explicitly enabled by the user since there are many customers with strict drafting standards who will not want to use this functionality.

Stretchable symbols: Select a single object such as polyline, rectangle, line or arc and click on the stretch icon in the symbol palette. Now click the mouse button next to a vertex or edge and stretch the drag image to the shape or size that you want. All properties stay attached to the symbol body.

Repeating Instances: These are a short hand form of FOR frame for use with a single instance. Instead of drawing a box around the instance with a FREXP property, the user gives the INST property a width similar to bus syntax, e.g. COMP(3:0). This defines the number of times that the instance COMP will be repeated and how the nets will be connected. If a bus connects into a bit wide pin on the repeated instance, the bus width must be the same as the repeat width, so that COMP#0 will get bit 0, COMP#1 will get bit 1, etc.

Chapter 2

Design Viewpoint Editor V8.5_1.8

Known Problems and Workarounds

This chapter lists the known problems in the Design Viewpoint Editor (DVE) V8.5_1.8, and describes workarounds where available.

- **DR 90939**

During export configuration, primitives are losing their property types

Description: If a primitive is defined with one of the values as a string, and another as an expression, the `$export_design_configuration()` command will only output one `$add_primitive()` command with a type of the last primitive added.

Workaround: None known.

Corrected Problems

Defects Corrected in This Release

The defects listed below have been corrected in this version of DVE.

- **DR 103683**

Latch Viewpoint fails on SDS designs

Description: Latching a viewpoint in DVE fails with an 'Illegal instruction' error on a design created by System Architect.

Resolution: The problem was that the System Architect references were circular, so in following them the code got into an infinite loop. The code now detects System Architect references and breaks the loop.

- **DR 93387**

Import back annotation fails with ‘yacc stack overflow’

Description: Because of a limitation in yacc, the parsing of a long ASCII BA file line on the HP-PA causes a ‘yacc stack overflow’ to occur.

Resolution: The yacc stack was defined as a static size on the HP/PA. It was changed to use a dynamic stack by changing a compile-time variable.

- **DR 77311**

Property is evaluated twice on lower level of hierarchy

Description: Top level instance, property width, value 10. Lower level instance, property width, value width*2.

When evaluated, bottom level property shows a value of 40. It should show 20.

Resolution: This was a problem only with the value displayed in a schematic source view. All other tools returned the correct value. The schematic source view has been changed to display the correct value.

- **DR 74203**

Limit on length of synonym when importing ASCII BA

Description: DVE generates a warning when importing an ASCII BA that has synonyms that exceed 80 characters in length and the back-annotation does not contain the synonym.

Resolution: This limit has been bumped up to 2048.

- **DR 117777**

W/ special char DVE can generate an ascii ba file it can't read

Description: If a property value has an embedded double quote character, and the property is exported into an ASCII back annotation file, the result will not

be correct. The entire property value needs to have quotes put around it, and the embedded quotes need to be escaped. So, a value of `first"second"third` will now be exported as `"first\"second\"third"`, which the import ASCII back annotation will correctly parse.

Resolution: The Export ASCII Back Annotation function will now correctly place back-slashes and quotes when needed.

Changes

The following section lists the functions that have been changed and the new functions. Each function in the changed section has one line which is **bold**. This is the argument that has been added. The functions in which the **bold** line is not the last line of the function are not transcript compatible with previous releases. Any custom userware that has been written to call those functions needs to be changed to account for the new argument. A full description of the functions listed can be found in the *Design Viewpoint Editor User's and Reference Manual*.

Changed Functions

```
$open_back_annotation(  
    ba_name : string,  
    read_only : switch name [noreadonly, readonly] {default = @noreadonly},  
    priority : integer {default = 0, check = priority >= 0}  
)
```

```
$add_back_annotation(  
    ba_name : string,  
    read_only : switch name [noreadonly, readonly] {default = @noreadonly},  
    priority : integer {default = 0, check = priority >= 0}  
)
```

```
$connect_back_annotation(  
    ba_name : string,  
    read_only : switch name [noreadonly, readonly] {default = @noreadonly},  
    priority : integer {default = 0, check = priority >= 0}  
)
```

```
$update_latched_version(  
    object_pathname: optional string,  
    type : switch name [all, sub_tree, component, back_annotation],  
    prune_depth : label integer,  
    revert: switch name [norevert, allow_revert, revert],  
    filter: rest  
)
```

```
$unlatch_version(  
    object_pathname: optional string,  
    type : switch name [all, sub_tree, component, back_annotation],  
    prune_depth : label integer,  
    force: switch name [noforce, force],  
    filter: rest  
)
```

New Function

```
$set_allow_fixed_ba(  
  allow_fixed : switch name [true, false] { default = @true }  
)
```

Enhancements

There have been no enhancements made other than those listed in the “Changes” section above.

Chapter 3

Conserve

V8.5_1.8

Known Problems and Workarounds

This chapter lists known problems in Conserve (DVAS) V8.5_1.8, and describes workarounds where available.

- **DR 106771**

Unselect -except fails with system defined properties

Description: The `$$unselect_except_by_property` fails with the system defined property “`$instance_name`”.

Workaround: None known.

- **DR 66978**

Incorrect display of buses in trace window

Description: A bus that is connected over a netcon or a frame to a second bus with the same name, but different direction, should be displayed in the trace window in the following two ways:

INBUS(0:7) and INBUS(7:0)

However, the trace window always shows just one of the buses.

Workaround: Rename one of the buses to match the other.

- **DR 67053**

add synonym doesn't work with pieces of a bus

Description: If a bus is defined as bus(31:0), adding a synonym for bus(3:0) will only add a synonym on the last bit.

Workaround: The problem is the object bus(3:0) does not exist, so there is nothing to create a synonym on. The workaround is to create a user defined bus for bus(3:0), then set the synonym to that.

Corrected Problems

Defects Corrected in This Release

The defects listed below have been corrected in this version of Conserve.

- **DR 108600**

\$get_typical() and \$get_worst_case(); Different return format.

Description: An instance has a triplet property value of "1 2 3". If you ask for the worst case value with \$get_worst_case(), you get the value "3". However, when you ask for the typical with \$get_typical(), you get the value "[2]".

Resolution: The \$get_best_case () and \$get_worst_case() functions were returning one number that was the maximum or minimum for all worst/best case properties on all objects. This was changed to work like typical, which returns the typical value for each selected object.

- **DR 96778**

Make monitor flag movable

Description: Monitor flags are automatically placed, so sometimes they need to be moved to a better location.

Resolution: It is now possible to move monitor flags once they have been placed by the application.

Changes

Conserve has been changed to allow access to Bundles, which were added in Design Architect. It is possible to select a bundle by just specifying the base name of the bundle (the part before the '{'), or by specifying the entire bundle name, including the members. Containment highlighting has also been updated so that a bundle will be containment highlighted when one of its members is selected, and a bundle member will be highlighted when any bundle it is a member of is selected.

Enhancements

There have been a set of new functions added to Conserve to allow configuration of monitor flags, and the creation of a “comment flag”. The following functions have been added to allow monitor flag attribute configuration (refer to the *Design Viewing and Analysis Support Manual* for details):

\$add_comment_flag()	\$set_default_monitor_flag_height()
\$delete_comment_flag()	\$get_monitor_flag_font()
\$get_comment_flags()	\$set_monitor_flag_font()
\$get_default_monitor_flag_font()	\$get_monitor_flag_height()
\$set_default_monitor_flag_font()	\$set_monitor_flag_height
\$get_default_monitor_flag_height()	

Chapter 4

Design File Interface

V8.5_1.8

Corrected Problems

Patches Included in This Release

The following patches have been incorporated into this release:

Patch p106

Patch p139

Patch p149

Defects Corrected in This Release

The defects listed below have been corrected in this version of DFI. If more than one defect report number is associated with a reported problem, the original defect report number is included in parentheses for reference.

- **DR 85215**

Description: No warning when user BA's to read-only back annotation object

Resolution: Added DFI will issue warning when opening a read-only BA and an error when making edits to a read-only BA.

- **DR 51224**

Description: Misleading error messages from DFI

Resolution: Previous versions of DFI only issued an error when trying to open a locked viewpoint or BA file when DFI `dfi_transcript_msg()` was set to `DFI_TRUE`. DFI has been modified so that the message will always be printed when trying to access a locked viewpoint.

- **DR 73244**

Description: DFI msg re. dynamic heap space is misleading

Resolution: DFI will no longer report “possibly due to bad viewpoint name or lack of dynamic heap space”.

- **DR 115602**

Description: When using the `rapid_scan` mode in DFI, the `inst_owner` field for nets and instances was not being filled in.

Resolution: The `inst_owner` field is now being filled in when using `rapid_scan`.

Enhancements

New DFI Functions

Please refer to *DFI User's and Reference Manual* for complete descriptions of these new functions.

`dfi_get_first_of_all_bundle_member()`

The `dfi_get_first_of_all_bundle_member()` routine starts a scan of all the members of a user defined bundle. If a member of this bundle is itself a bundle this routine may be called again (with a different bundle record) to scan the members of the sub-bundle. If you do not wish to scan all the members of the bundle, you can terminate the scan at any time by calling `dfi_quit_bundle_member_scan()`.

dfi_get_next_bundle_member()

The `dfi_get_next_bundle_member()` routine returns the bundle (net) record for the next member of the current user defined bundle member scan corresponding to the input bundle (net) record identifier parameter.

dfi_quit_bundle_member_scan()

The `dfi_quit_bundle_member_scan()` routine terminates the bundle member scan corresponding to the input bundle (net) record parameter. This function should always return a NULL pointer to signify the scan has indeed terminated.

dfi_is_net_bundle()

The `dfi_is_net_bundle()` routine allows users to determine if a given net is a user defined bundle.

dfi_is_property_fixed()

The `dfi_is_property_fixed()` routine allows users to determine if a given property is fixed.

Chapter 5

QuickCheck V8.5_1.8

Known Problems

Name Checker — In past releases, name checker annotated a name translation to all equivalent nets in a design. Some of these equivalent nets were rippers (R\$). In B.1 rippers no longer exist in the evaluated fraction and therefore do not need name translation. Also, since they do not exist in the evaluated model, they can not be annotated. Existing Back Annotation files which contain annotations to rippers will now list the rippers in the “Invalid Objects” section of the BA window, and will cause Errors and Warnings when trying to export the Back Annotation. If an ASCII BA file is read which has rippers listed, the rippers will not be found, leading to Errors and Warnings. To remove the references to the rippers remove the invalid objects by issuing “\$delete_invalid_entries()”.

QuickCheck — If a NULL string is passed to an OUTPUT statement, a crash will result when running on a Solaris machine. All other platforms will handle this situation and not crash.

Corrected Problems

Defects Corrected in This Release

The defects listed below have been corrected in this version of QuickCheck. If more than one defect report number is associated with a reported problem, the original defect report number is included in parentheses for reference.

- **DR 83006**

Description: Change Queue not being reset after fail.

Resolution: QuickCheck has been modified to reset the change queue after failure.

- **DR 93090**

Description: Global net considered not global.

Resolution: QuickCheck has been modified to check each equivalent net determine if any are global.

- **DR 63326**

Description: String length limitation.

Resolution: QuickCheck has been modified to accept strings up to 8192 characters.

- **DR 93091**

Description: DMGR invocation of config_erc fails.

Resolution: QuickCheck has been modified to correctly invoke from DMGR.

Changes and Enhancements

Other than the “Corrected Problems” discussed earlier in this chapter, this release of QuickCheck contains no changes or enhancements.

Chapter 6

CDP

V8.5_1

Known Problems and Workarounds

- **DR Item 114544**

CDP accesses other than the specified viewpoint

Description: The problem comes from a call "cdpdrCellFindDesignConfigByName" in the miflist program. That call says that the entire contents of the cell should be searched for a "design config" object. That, as a side-effect, causes the entire contents of the cell to be read into memory, including the "dummy" viewpoint. (The Eddm object "Eddm_design_viewpoint" maps to a "design config" in the CFI data model).

Since the dummy viewpoint isn't complete, reading it causes error messages to be generated. Those messages aren't "fatal" because they don't affect any of the other objects in memory.

Corrected Problems

Defects Corrected in This Release

- **DR Item 110300**

CDP scans wrong connectivity due to repeated hierarchy

Description: This design had a piece of hierarchy that was used at multiple places in the design. The parent of these places had the characteristic that its

connectivity was dependent on one of the same properties (but a different value) that the hierarchy underneath it was dependent on.

This property "clash" where the connectivity dependent property list of a section of the design was dependent on two different values of the same property, wasn't being handled correctly. This case is now checked for and handled.

Workaround: The workaround to this problem is to not use repeated hierarchy. The customer should make a copy of the SSEL block and use the copied block in place of the repeated instance.

- **DR Item 115627**

Incorrect property value expression

Description: CDP failed to return the correct value for an expression property on a primitive instance.

- **DR Item 114402**

CDP did not evaluate properties (with -eval or -evalall)

Description: CDP failed to evaluate expression properties.

- In addition, the following defects were fixed by patches included in B.1:

DR Item	Description	Patch #
108670	cfidrCellGetViews() fails on BAR00 of TI	p193
109962	CDP can't get flat net attached to port	p204
105163	expressions with conditional operators a	p172
114334	can't get flat net attached to port	p221

Chapter 7

EDIF (ENRead, ENWrite)

V8.5_1.1

Known Problems and Workarounds

This section lists known problems in ENRead and ENWrite V8.5_1.1, and describes workarounds where available. If more than one defect report number is associated with a reported problem, the numbers of duplicate defect reports follow the original defect report number in parentheses. If no numbers are associated with certain defects, those defects are listed in alphabetical order at the end of the numerical listing.

ENRead

- None

ENWrite

- **DR 87663**
Setup Character Substitution -Names crashes.

Description: The “Setup Character Substitution -Names” command crashes on trivial designs. Note that it is only the -Names argument, and not the -Identifiers argument, that results in a crash.

Workaround: The default for ENWrite’s setup character substitution command is -identifiers, and this should be used as the workaround when needed.

- If the Setup Library Mode -Split command is used with the Setup Name Property -Cell command, the library names will contain the original cell name, and not the cell name based on Setup Name Property -Cell. For example, instance I\$1 references component \$LSI_LIB/non_prim_part. If I\$1 contained a “comp” property with value “74ls00”, then the library name created (when Setup Library Mode -Split is used) *should* be “\$LSI_LIB__74ls00”, but what is currently generated is a library named “\$LSI_LIB__non_prim_part”.

Corrected Problems

Patches Included in This Release

The following patches have been incorporated into this release:

Patch p108

Patch p148

Defects Corrected in This Release

The defects listed below have been corrected in this version of ENRead and ENWrite. If more than one defect report number is associated with a reported problem, the original defect report number is included in parentheses for reference.

ENRead

- None

ENWrite

- **DR 83679**
Would like to have Setup Name Property -Cell

Description: A MGC customer cannot use EDIF to complete their design flow because their tool requires that the EDIF netlist contains cell names in upper

case. ENWrite, however, writes them out in whichever case the cell name is defined in the Mentor system (which is lower case).

Resolution: The setup name property command has been extended as follows:

Setup Name Property “<list of property names>” -Cell

When this command is used and a special property exists on the symbol or the instance, ENWrite will use the value of this property as the cell name. When more than one property is given to the command, the properties must be included in double quotes, as in:

Setup Name Property “comp mcomp model” -Cell

If only one property is to be used with the command, the quotes are not necessary, as in:

Setup Name Property Comp -Cell

- **DR 75687**

Library gets referenced before it is declared

Description: In some designs, ENWrite creates an EDIF netlist with the wrong order of library declarations and references. A number of libraries get referenced before they are actually declared.

Resolution: ENWrite has been updated to correctly detect and correct forward references.

- **DR 95283 (DR 75687)**

Output EDIF netlist from DFT can't be read into FastScan

Description: In some designs, ENWrite creates an EDIF netlist with the wrong order of library declarations and references. A number of libraries get referenced before they are actually declared.

Resolution: ENWrite has been updated to correctly detect and correct forward references.

Changes

ENWrite no longer outputs the “pintype” property on nets. The “pintype” property on a net is used by ENWrite to determine port direction information when the interface is not defined. The “pintype” property is transferred to the port definition and is not needed in the net definition.

Enhancements

ENWrite has been updated so it can create an EDIF netlist from designs containing Design Architect net and pin bundles. ENWrite ignores the bundle information and does not transfer it to the EDIF netlist.