

# Drawing Generation Best Practices

---

CADD5® 5 15.0

DOC40201-005

---

**Copyright © 2007 Parametric Technology Corporation. All Rights Reserved.**

User and training guides and related documentation from Parametric Technology Corporation and its subsidiary companies (collectively "PTC") is subject to the copyright laws of the United States and other countries and is provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes.

Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

**UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.**

For Important Copyright, Trademark, Patent, and Licensing Information: For Windchill products, select About Windchill at the bottom of the product page. For InterComm products, on the Help main page, click the link for Copyright 2007. For other products, select Help > About on the main menu for the product.

**UNITED STATES GOVERNMENT RESTRICTED RIGHTS LEGEND**

This document and the software described herein are Commercial Computer Documentation and Software, pursuant to FAR 12.212(a)-(b) (OCT'95) or DFARS 227.7202-1(a) and 227.7202-3(a) (JUN'95), and are provided to the US Government under a limited commercial license only. For procurements predating the above clauses, use, duplication, or disclosure by the Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 (OCT'88) or Commercial Computer Software-Restricted Rights at FAR 52.227-19(c)(1)-(2) (JUN'87), as applicable. 02202007

**Parametric Technology Corporation, 140 Kendrick Street, Needham, MA 02494 USA**

---

---

# Table of Contents

---

## Preface

Related Documents .....	vii
Book Conventions .....	vii
Window Managers and the User Interface .....	viii
Online User Documentation .....	viii
Online Command Help .....	ix
Printing Documentation .....	x
Resources and Services .....	x
Documentation Comments .....	x

## Overview

Introduction .....	1-2
The Drawing Generation Process .....	1-3
Creating Drawings .....	1-3
Updating the Drawing After Changes to the Parametric Model ..	1-4
Release 10 Onward .....	1-5
Revision 8 Specific .....	1-5

## Creating and Updating Part Detail Drawings

Creating Detailed Drawings for Parts .....	2-2
Setting Up CADDs for Detailing .....	2-3
Setting Up Fonts .....	2-3

---

Setting Up the Parameter File _____	2-3
Creating New Explicit Drawing Forms _____	2-4
Annotating the Title Blocks Using Tnodes _____	2-5
Suggested Customization to Reduce Menu Selections _____	2-6
Revision 9 Onward _____	2-6
Revision 8 Specific _____	2-6
Configuring Memory Requirement for HLR _____	2-7
Temporary File Space _____	2-7
Errors for Insufficient Space _____	2-7
Checking the Validity of Components _____	2-8
Activating a Part _____	2-10
Activating a New Drawing _____	2-11
Creating Drawing Layouts _____	2-12
Setting Up a General Drawing Layout _____	2-12
Aligning Two Views _____	2-13
Moving Views to Change the Layout _____	2-16
Moving a View on the Drawing _____	2-16
Moving All Views on the Drawing _____	2-17
Creating Detail or Scrap Views _____	2-17
Creating Section Views _____	2-18
Using the DEFINE SECTION Command _____	2-18
Advantages _____	2-18
Capabilities _____	2-18
Creating a Section in a Separate View _____	2-21
Creating a Simple Section View _____	2-22
Viewing Sectioning Results for Revision 8 and 9 _____	2-22
Creating an Unfolded Section _____	2-22
Creating a Section in an Existing View _____	2-23
Points to Note When Creating Sections _____	2-25
Updating a Section View _____	2-26
Rules for Updating a Section View _____	2-27
Creating Centerlines _____	2-28
Using the Conventional and Parametric Methods _____	2-28
Conventional Method _____	2-28

---

Parametric Method _____	2-28
Defining Features to Create Centerlines _____	2-29
Basic User-Defined Features _____	2-29
Advanced User-Defined Features _____	2-30
Creating HLR Views _____	2-32
Example of an HLR View _____	2-32
Ensuring Quality of the HLR Image _____	2-33
Using the Faceted Engine _____	2-34
Using the Analytical Engine _____	2-35
Updating an HLR View _____	2-35
Updating an HLR View from Release 11 Onward _____	2-36
Unhiding the HLR View _____	2-36
Layering Conventions and HLR _____	2-37
Adding Dimensions _____	2-39
Creating Explicit Dimensions _____	2-39
Associative Dimensions _____	2-39
Tips for Creating Dimensions _____	2-40
Using Design Dimensions — A Parametric Approach _____	2-41
Advantages _____	2-41
Disadvantages _____	2-41
Creating Design Dimensions _____	2-42
Managing Design Dimensions _____	2-44
Solving Design Dimensioning Issues _____	2-47
Filing the Part for Saving the Drawing _____	2-48
Using the SAVE SETUP Command _____	2-48
Using the RESTORE SETUP Command _____	2-49
Updating Detail Drawings for Parts _____	2-50
Updating a Drawing for Minor Modeling Changes _____	2-50
Updating a Drawing for Major Modeling Changes _____	2-51

## Creating and Updating Assembly Detail Drawings

Considerations for Assembly Drawing Creation _____	3-2
Points to Note Before Activating an Assembly _____	3-3
Performing Database Checks _____	3-3

---

Updating Drawings Using the update_fig Script _____	3-4
Updating Drawings Using the update_gr Script _____	3-4
Creating Assembly Detail Drawings _____	3-5
Setting Up CADDs _____	3-7
Activating an Assembly _____	3-8
Activating an Adrawing _____	3-8
Working with Large Assemblies _____	3-9
Activating a Drawing in the Explicit Environment _____	3-10
Copying Similar Drawings _____	3-10
Defining Required Views _____	3-11
Creating Section Views for Assembly Drawings _____	3-12
Creating HLR Views on Assembly Drawings _____	3-13
Working with HLR Views _____	3-13
Creating Exploded HLR Views _____	3-13
Validating an Assembly _____	3-14
Adding Dimensions, Centerlines, or Annotations to Assembly Drawings _____	3-15
Saving the Setup and Filing the Adrawing _____	3-16
Updating the Assembly Detail Drawing _____	3-17
Updating the Drawing for Minor Changes in the Assembly _____	3-17
Cleaning the Database _____	3-18
Viewing the Components _____	3-18
Updating Section Views _____	3-18
Updating HLR Views _____	3-19
Updating the Drawing for Major Changes in the Assembly _____	3-19

## Enhancements in Drawing Generation

Revision-Specific Enhancements _____	A-2
--------------------------------------	-----

---

# Preface

---

*Drawing Generation Best Practices* is intended to help CADDs users improve their skills in creating and updating piece part and assembly drawings. The general drawing creation method is based on the enhancements that were developed for various CADDs revisions, beginning with Revision 8.

## Related Documents

The following documents may be helpful as you use *Drawing Generation Best Practices*:

- *Explicit Modeling User Guide and Menu Reference*
- *Appearance and Hidden Line Removal Reference*
- *Hidden Line Removal (and AEC HLR) User Guide and Menu Reference*
- *Sectioning User Guide*
- *Design and Drafting User Guide and Menu Reference*
- *Parametric Modeling User Guide and Menu Reference*
- *Concurrent Assembly Mock-Up Best Practices*

## Book Conventions

The following table illustrates and explains conventions used in writing about CADDs applications.

<b>Convention</b>	<b>Example</b>	<b>Explanation</b>
Menu selections and options	List Section option, Specify Layer field	Indicates a selection you must make from a menu or property sheet or a text field that you must fill in.
User-selected graphic location	X, d <sub>1</sub> or P1	Marks a location or entity selection in graphic examples.

Convention	Example	Explanation
User input in CADDs text fields and on any command line	<code>cvaec.hd.data.param</code> <code>tar -xvf /dev/rst0</code>	Enter the text in a CADDs text field or on any command line.
System output	<code>Binary transfer complete.</code>	Indicates system responses in the CADDs text window or on any command line.
Variable in user input	<code>tar -cvf /dev/rst0 filename</code>	Replace the variable with an appropriate substitute; for example, replace filename with an actual file name.
Variable in text	<code>tagname</code>	Indicates a variable that requires an appropriate substitute when used in a real operation; for example, replace tagname with an actual tag name.
CADDs commands and modifiers	<code>INSERT LINE TANTO</code>	Shows CADDs commands and modifiers as they appear in the command line interface.
Text string	<code>"SRFGROUPA" or 'SRFGROUPA'</code>	Shows text strings. You must enclose text string with single or double quotation marks.
Integer	<code>n</code>	Supply an integer for the <i>n</i> .
Real number	<code>x</code>	Supply a real number for the <i>x</i> .
#	<code># mkdir /cdrom</code>	Indicates the root (superuser) prompt on command lines.
%	<code>% rlogin remote_system_name -l root</code>	Indicates the C shell prompt on command lines.
\$	<code>\$ rlogin remote_system_name -l root</code>	Indicates the Bourne shell prompt on command lines.

## Window Managers and the User Interface

According to the window manager that you use, the look and feel of the user interface in CADDs can change. Refer to the following table:

### Look and Feel of User Interface Elements

User Interface Element	Common Desktop Environment (CDE) on Solaris and HP	Window Manager Other Than CDE on Solaris, HP, and Windows
Option button	ON — Round, filled in the center OFF — Round, empty	ON — Diamond, filled OFF — Diamond, empty
Toggle key	ON — Square with a check mark OFF — Square, empty	ON — Square, filled OFF — Square, empty

## Online User Documentation

Online documentation for each book is provided in HTML if the documentation CD-ROM is installed. You can view the online documentation in the following ways:

- From an HTML browser
- From the Information Access button on the CADDSS desktop or the Local Data Manager (LDM)

Please note: The LDM is valid only for standalone CADDSS.

You can also view the online documentation directly from the CD-ROM without installing it.

From an HTML Browser:

1. Navigate to the directory where the documents are installed. For example,  
/usr/apl/caddss/data/html/htmldoc/ (UNIX)  
Drive:\usr\apl\caddss\data\html\htmldoc\ (Windows)
2. Click `mainmenu.html`. A list of available CADDSS documentation appears.
3. Click the book title you want to view.

From the Information Access Button on the CADDSS Desktop or LDM:

1. Start CADDSS.
2. Choose Information Access, the `i` button, in the top-left corner of the CADDSS desktop or the LDM.
3. Choose DOCUMENTATION. A list of available CADDSS documentation appears.
4. Click the book title you want to view.

From the Documentation CD-ROM:

1. Mount the documentation CD-ROM.
2. Point your browser to:  
CDROM\_mount\_point/htmldoc/mainmenu.html (UNIX)  
CDROM\_Drive:\htmldoc\mainmenu.html (Windows)

## Online Command Help

You can view the online command help directly from the CADDSS desktop in the following ways:

- From the Information Access button on the CADDSS desktop or the LDM
- From the command line

From the Information Access Button on the CADDs Desktop or LDM:

1. Start CADDs.
2. Choose Information Access, the *i* button, in the top-left corner of the CADDs desktop or the LDM.
3. Choose COMMAND HELP. The Command Help property sheet opens displaying a list of verb-noun combinations of commands.

From the Command Line: Type the exclamation mark (!) to display online documentation before typing the verb-noun combination as follows:

```
#01#!INSERT LINE
```

## Printing Documentation

A PDF (Portable Document Format) file is included on the CD-ROM for each online book. See the first page of each online book for the document number referenced in the PDF file name. Check with your system administrator if you need more information.

You must have Acrobat Reader installed to view and print PDF files.

The default documentation directories are:

- `/usr/apl/cadds/data/html/pdf/doc_number.pdf` (UNIX)
- `CDROM_Drive:\usr\apl\cadds\data\html\pdf\doc_number.pdf` (Windows)

## Resources and Services

For resources and services to help you with PTC (Parametric Technology Corporation) software products, see the *PTC Customer Service Guide*. It includes instructions for using the World Wide Web or fax transmissions for customer support.

## Documentation Comments

PTC welcomes your suggestions and comments. You can send feedback electronically to `doc-webhelp@ptc.com`.

This chapter gives an overview of drawing generation best practices.

- Introduction
- The Drawing Generation Process

# Introduction

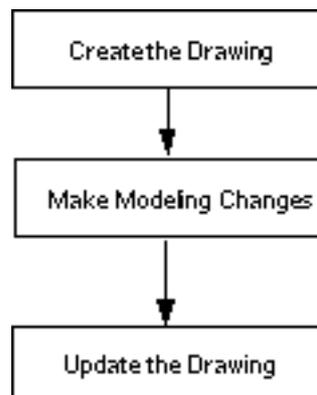
This chapter gives an overview of the drawing generation process for CADDs Revision 8 onward. The next chapters describe the best methods for creating and updating drawings for parts and assemblies. Information that is specific to a release is mentioned separately within each section.

In CADDs, detailed drawings are created in the Explicit environment using the drafting and detailing tools. The drawings created in the Explicit environment are associated with the model. Therefore, the HLR views, section views, dimensions, and the annotations are associative. Whenever you make changes to the model, update the drawing to reflect the changes in the model. To update the drawing, you must regenerate the HLR and section views.

It is also imperative that you maintain a clean database. To do this you must keep the explicit drawings and the parametric model synchronized by first updating the HLR and section views in the drawing and then correcting the dimensions.

# The Drawing Generation Process

This section provides a flowchart that describes the basic process of drawing generation. The drawing generation process illustrated in the flowchart is common to both the parts and the assemblies. Each of the steps shown in the flowchart have a number of subprocesses that are different for parts and assemblies.



## Creating Drawings

To achieve the best results, you must follow these guidelines when creating drawings:

- Before creating a drawing, check the integrity of the model. You can check the integrity of the parametric model by changing a parameter and regenerating the model.
- It is imperative that you follow good working practices when you create the initial drawing. These practices are described in the following section.

The general process of creating drawings is similar for both parts and assemblies. For the details related to the process of creating drawings for parts, see Chapter 2, “Creating and Updating Part Detail Drawings.” For details on creating drawings for assemblies, see Chapter 3, “Creating and Updating Assembly Detail Drawings.” The general drawing generation process is as follows:

1. Check the integrity of the parametric model.
2. File the part in the Explicit environment.
3. Use CHECK DBASE in the Explicit environment to check the validity of the part.
4. Define the views and drawing layouts for the drawing and set the required views.
5. Create section views.

6. Add centerlines.
7. Create the HLR view for the views as required.
8. Add dimensions and annotations.
9. Use CHECK DBASE from the Explicit environment to check the model.
10. File the part using FILE PART in the Explicit environment.

Release 13 onwards, filing a part using FILE PART in the Parametric environment updates the part in Explicit environment also.

Please note: Filing the part in the Explicit environment ensures that the part does not cause problems when it is used as a component in an assembly.

## Updating the Drawing After Changes to the Parametric Model

Synchronizing the explicit drawing with the parametric model helps in maintaining a clean database. Always update the drawings when changes are made to the model in the Parametric environment. Updating the drawing requires updating the HLR and section views and the dimensions.

Release 10 Onward : Reassociate the marked dimensions.

Revision 8 and 9 Specific: Recreate the dimensions that may be lost due to changes to the model.

The general process of updating the drawing after the parametric model has undergone changes is as follows:

1. Check the integrity of the parametric model after making the modeling changes.
2. Enter the Explicit environment.
3. Use CHECK DBASE in the Explicit environment to check the model.
4. Use FILE PART to file the part after the part has been successfully checked.
5. Update the HLR and section views.
6. Reassociate the dimensions and correct the dimensioning if necessary.
7. Use CHECK DBASE from the Explicit environment to check the model.
8. File the part.

## Release 10 Onward

The efficiency of the drawing update process since CADDs Release 10 is due to the following features:

- Before updating the HLR and section views, you can query for information on the entire drawing for views that require an update. Update only those views that require the update.
  - To query details on the entire drawing, choose the Verify (?) option on the top bar menu. Choose the Drawing option from the resultant menu. The Drawing Data menu appears with all the information related to the drawing displayed in the menu window. You can also make changes to the drawing using the Drawing Data menu.
  - To query the details on any view, choose the Verify (?) option available on the top bar menu. Choose the View Data option from the resultant menu. The View Data menu appears with all the information related to the drawing displayed in the menu window. You can also make changes to the view using the View Data menu.
- You are automatically prompted when a drawing requires an update. The section views and HLR views that require an update due to the modeling changes are highlighted in a different color. The dimensions that are associated to changed entities when regenerated are highlighted in a different color.

Please note: Set the `UPDATE_DUE_VIEW_COLOR` environment variable to the required color in the `.caddsrc-local` file. The changed views are highlighted in the specified color.

## Revision 8 Specific

The following points are specific to Revision 8.

- You must use `UNDEFINE SECTION` to remove any section views that have been created using `DEFINE SECTION CUTAWAY`.
- Use `REGENERATE SECTION ALL` to update the rest of the section views on the drawing.



# Creating and Updating Part Detail Drawings

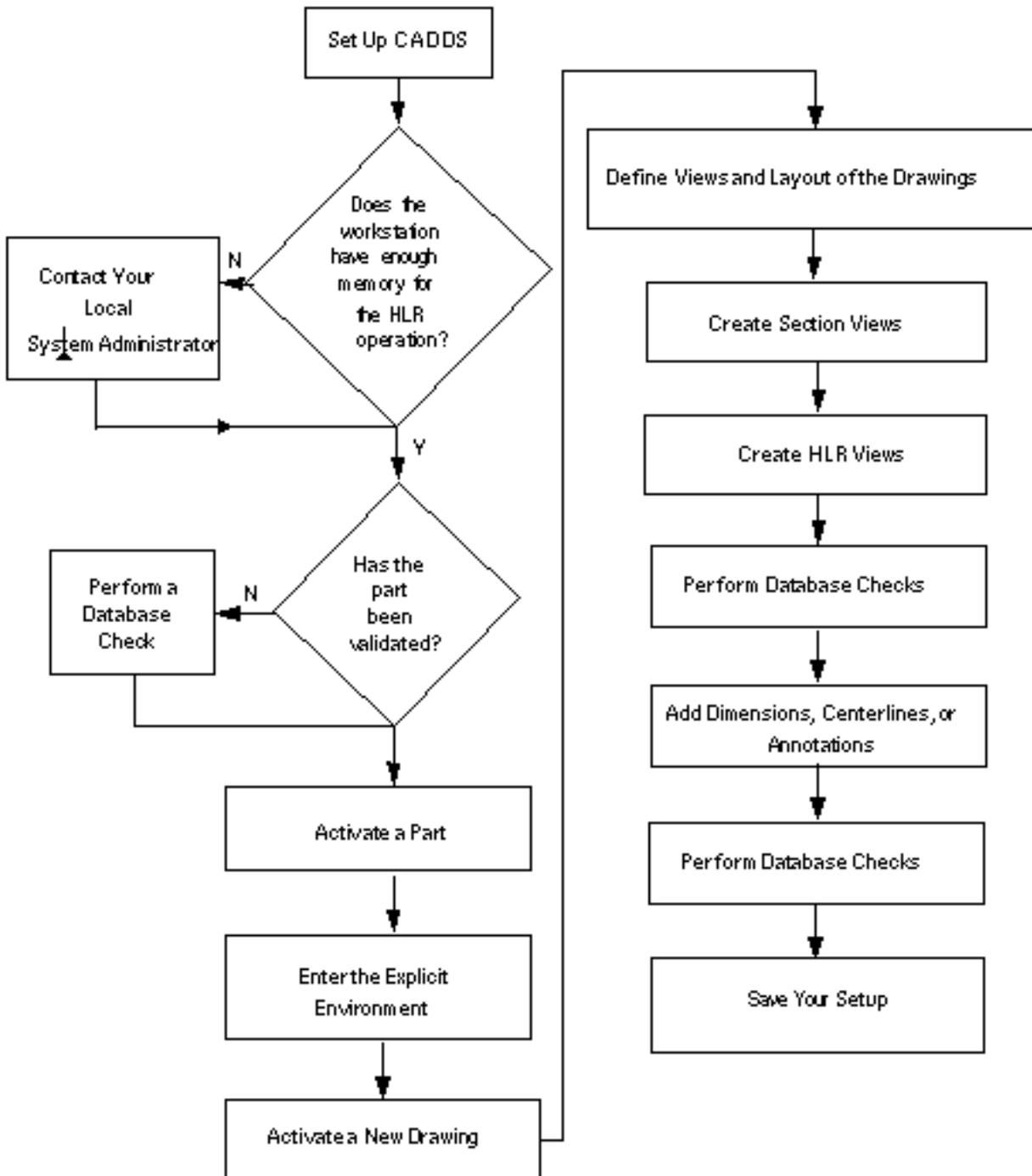
---

This chapter describes the best practices for creating and updating detailed drawings for parts.

- Creating Detailed Drawings for Parts
- Setting Up CADDs for Detailing
- Configuring Memory Requirement for HLR
- Checking the Validity of Components
- Activating a Part
- Activating a New Drawing
- Creating Drawing Layouts
- Creating Section Views
- Creating Centerlines
- Creating HLR Views
- Layering Conventions and HLR
- Adding Dimensions
- Filing the Part for Saving the Drawing
- Updating Detail Drawings for Parts

# Creating Detailed Drawings for Parts

The following flowchart illustrates the best process for creating detailed drawings for parts. The details of each step in the flowchart are described in the relevant sections of this chapter.



# Setting Up CADDs for Detailing

CADDs requires specific settings to achieve the best results during the detailing process. These settings are described in the following sections.

## Setting Up Fonts

A standard compiled version of the standard imperial font definition set is included with CADDs. Run the following commands as a root user to set the metric fonts:

```
% source .caddsrc  
% /usr/apl/cadd/bin/Fontdefs  
=usr.apl.cadd.data.fontdefs-mm
```

## Setting Up the Parameter File

The explicit parameter file allows you to set CADDs to suit the requirements of your local or international dimensioning standards. This file also enables you to control certain behavior such as associativity. To maintain a fully associative drawing, set the SELECT DIMENSION command as follows:

```
Select Dimension Retain ON
```

The following settings help to improve the usability and placements of dimensions when using design dimensions:

```
Select Dimension Text Location Manual  
Select Dimension Nocheck
```

The SELECT DIMENSION RETAIN ON command ensures that no dimensions are deleted from the drawing. The dimensions that you cannot associate with the changed geometry after making modeling changes are made nonassociative and are flagged automatically. These nonassociative flagged dimensions are retained on the drawing. The change in dimension is indicated by the following color code:

- Red indicates the dimensions dissociated or marked for deletion or reassociation
- Blue indicates the dimensions whose values have changed
- Green indicates the dimensions whose values are the same but the position is changed

Dimensions that have not changed are displayed with their original attributes.

To change the settings in the parameter file, perform the following steps:

1. Enter the Explicit environment.
2. Make changes to the setup as described.
3. Use `SAVE PARAMETER <name of the new parameter file>`. A new binary parameter file is created.
4. Set the following environment variable in the `.caddsrc-local` file so that CADDs can use the parameter file.

```
setenv CADDSPARAM <name of the parameter file>
```

Please note: You can use the same explicit parameter file for both UNIX and Windows. Run the `/usr/apl/cadd/bin/CONVERT` executable for one-time conversion of the explicit parameter files created on Windows with the earlier versions of CADDs, to the default UNIX format. You can run this executable as shown below:

```
CONVERT <InputFile> <OutputFile>
```

where:

InputFile: The legacy parameter file to be converted.

OutputFile: The output parameter file that is created after conversion.

Revisions 8 and 9 Specific: You must manually set the following parameters in the parameter file:

```
Select Dimension Retain ON  
Select Dimension Nocheck
```

Please note: These settings are the default from CADDs Release 10 onward.

## Creating New Explicit Drawing Forms

The best process to create a new explicit drawing form follows. The next section describes annotating title blocks and customizing menus.

To create a new explicit drawing form and include it in the new drawing list, use the double-precision format for the CADDs databases.

1. Select `pformat double`. The standard part format is C5. The double-precision format ensures that CADDs does not create an `_fd` file. If the part has been created in C5 format, use `EXTRACT PART` to remove the `_fd` file.
2. Create an explicit part with a drawing. Use a proper naming convention. For example, `draw1`.

3. Create a drawing border for the drawing in the Draw mode. Define the standard views for the drawing.

4. Use the following command to create a Nodal figure for the part:

```
EXIT PART FILE Nfig <name of the part> draw <name of the drawing, in this  
case, draw1> All
```

5. Use UNIX commands to remove any `vp_links` directories from the part directory and place the part in the following directory:

```
/usr/apl/cadds/data/form
```

The new explicit drawing form automatically appears in the list.

## Annotating the Title Blocks Using Tnodes

Use tnodes to create the title blocks for the drawing forms and then annotate the title block using ANNOTATE NFIGURE. To use ANNOTATE NFIGURE, add a property called AUTOANN to the tnode. The value of AUTOANN is the prompt that you get when using ANNOTATE NFIGURE. Use a tnode to represent every text field. The tnode holds the following attributes for the text:

- Size
- Font
- Justification

When you use ANNOTATE NFIGURE, you are prompted to enter the title on the command line. ANNOTATE NFIGURE prompts you for tnode values in the order in which they are created.

The following is a sequential list of commands that you can use to annotate the title blocks:

```
##  
Insert tnode  
Insert Property AUTOANN <enter the title in double quotes>:  
draw/model ent <select the Tnode>  
Construct nfigure <name>  
Insert nfigure <name>  
annotate nfigure <pick the figure>
```

Please note: To use the drawing forms and their title blocks effectively and efficiently, you must follow the annotation process properly.

## Suggested Customization to Reduce Menu Selections

Customizing menus enables you to reduce the number of menu selections when creating a detailed drawing. Use the Customizer to customize the menus. The following is a list of revision-specific customizations:

### Revision 9 Onward

The following is a list of customizations for Revision 9 onward:

```
CHANGE DIMENSION ALIGN (align two dimensions)
CHANGE DIMENSION ASSOCIATIVE (change an existing nonassociative
dimension to associative)
ERASE ENTITY EDGE (blank the edge of a solid for cleaning up HLR
views)
```

### Revision 8 Specific

The following is a list of customizations for Revision 8:

```
INSERT LDIM HORIZONTAL (create a horizontal linear dimension)
INSERT LDIM VERTICAL (create a vertical linear dimension)
CHANGE DIMENSION ALIGN (align two dimensions)
CHANGE DIMENSION ASSOCIATIVE (changing an existing nonassociative
dimension to associative)
ERASE ENTITY EDGE (blank the edge of a solid for cleaning up HLR
views)
```

# Configuring Memory Requirement for HLR

The HLR process requires a specific amount of memory that must be proportional to the size of the assembly or part to successfully perform the requested operation.

The required memory is configured on each workstation in the form of swap space or temporary file space.

## Temporary File Space

The temporary file space contains all temporary files created by CADDs and various system processes. For example, on UNIX systems these files are stored in the `/tmp` and `/usr/tmp` directories. When performing HLR tasks, CADDs creates temporary files as it processes the HLR request and stores them in the temporary file space provided on the workstation.

Please note: Configure the `/usr/tmp` directory so that it is cleaned out every time you exit CADDs.

## Errors for Insufficient Space

If the temporary file space contains data before you request an HLR operation, then CADDs may have insufficient space for creating files. If the assembly or part is very large, then the memory may be insufficient. The insufficient memory causes CADDs to fail during an HLR process. You may get the following error messages:

- Segmentation or Bus error in the case of insufficient local disk space.
- Malloc, Alloc, Realloc, and memory errors in the case of insufficient swap space on the disk.

Please note: If you encounter one or both of these errors, contact your local system administrator. The administrator can check for the memory allocation on your workstation and remove any unwanted files or reconfigure your disk space.

## Checking the Validity of Components

You must perform database checks on parts that have originated from other CAD systems and have been converted to CADDs parts using the translator. These components can contain large portions of wasted space in the part database, which can result in numerous errors in CADDs. The errors may be system errors, such as segmentation violations or malloc errors. These errors can manifest as components that cannot be excluded by layers or can produce poor HLR results in the assembly environment.

To perform database checks, use one of the following methods:

- Use CHECK DBASE in the Explicit environment and file the part.
- Use the ckCAD utility provided by CADtools.

Release 13 Onward: You can use the `validate_db` utility for automatic database validation when activating or filing a part. To use this utility, set the following environment variables in the `.caddsrc` file to the path of the `validate_db` executable file:

- `DB_CHECKING`
- `DB_CHECKING_ACT`
- `DB_CHECKING_FILE`

Setting the `DB_CHECKING_ACT` environment variable in the `.caddsrc` file enables database validation when activating a part. Setting the `DB_CHECKING_FILE` environment variable enables database validation when filing a part.

If you have set the `DB_CHECKING` environment variable to the path of the `validate_db` executable file, you can use the ckCAD utility for automatic database validation when activating or filing a part. If you have not set the `DB_CHECKING` environment variable, the `validate_db` utility is used for automatic database validation.

Please note: Ensure that the relevant checks have been performed on the part before filing the part, and ensure that only clean parts are stored in the Vault.

In case of doubt, contact customer support. For details see the section “Resources and Services” on page -x.

### Warning

The commands or techniques to clean and validate parts that have not been described in this document are not supported by PTC.

Please note: If components on the CAMU tree are not in a clean state, that is, they have not had a database management routine run on them successfully, then the HLR process may not produce the required results.

## Activating a Part

Activate a part only after ensuring that the part has been validated. For details on validating a part, see the section “Checking the Validity of Components” on page 2-8. If you have activated a part that has not yet been validated, first validate it using CHECK DBASE. You can then enter the Explicit environment and activate a new drawing.

## Activating a New Drawing

To activate a new drawing:

1. Enter the Explicit environment.
2. Choose the Activate New Drawing option from the Drawing menu on the top bar. The Activate New Drawing menu appears.
3. Select the required drawing sheet from the list that appears on the Activate New Drawing menu. Specify a name for the drawing in the Drawing field.
4. Click Apply to display a new drawing sheet.

## Creating Drawing Layouts

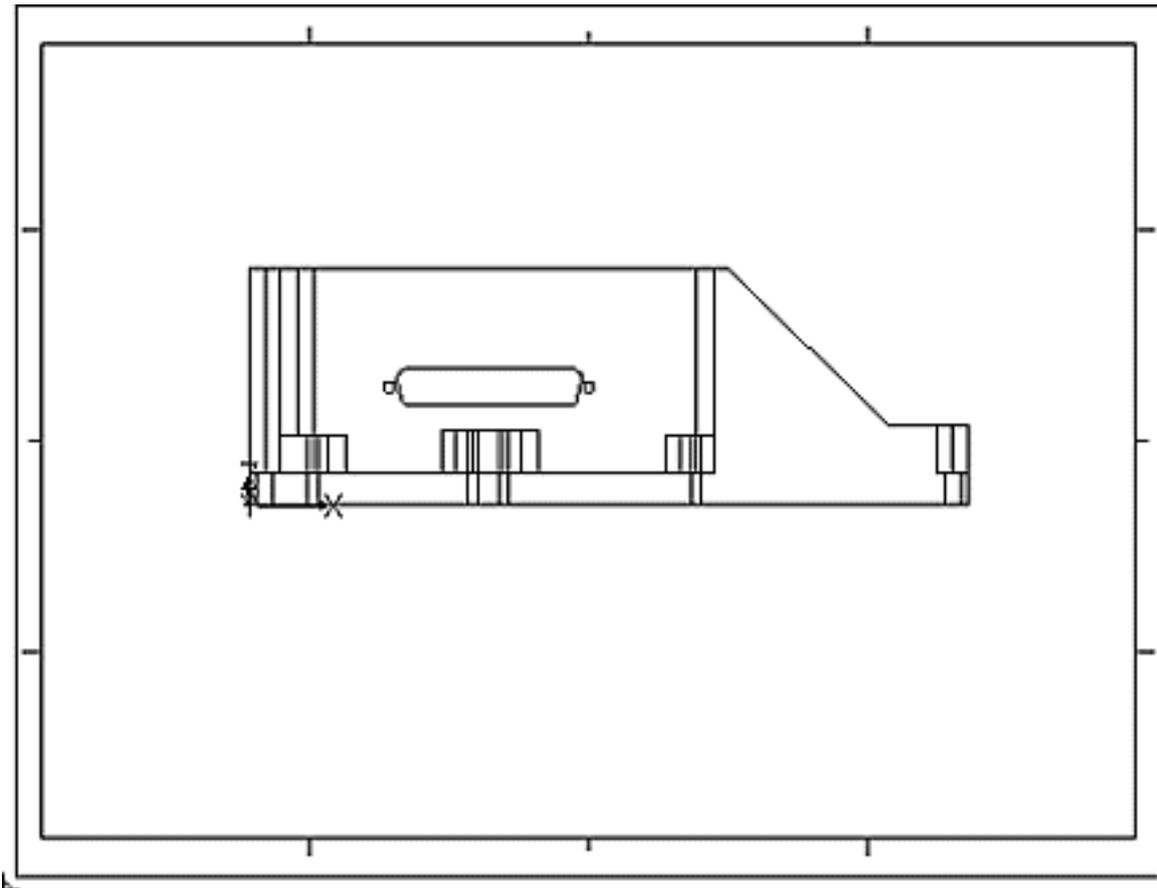
There are various methods to lay out a drawing. The best method for defining views and creating the layout of the drawing under construction is described in the following sections.

### Setting Up a General Drawing Layout

To set up the layout of a drawing, use this general drawing layout method:

1. Define a view with the correct location and view scale factors. This view is a reference view for views that are created later.

**Figure 2-1 Define the First View**



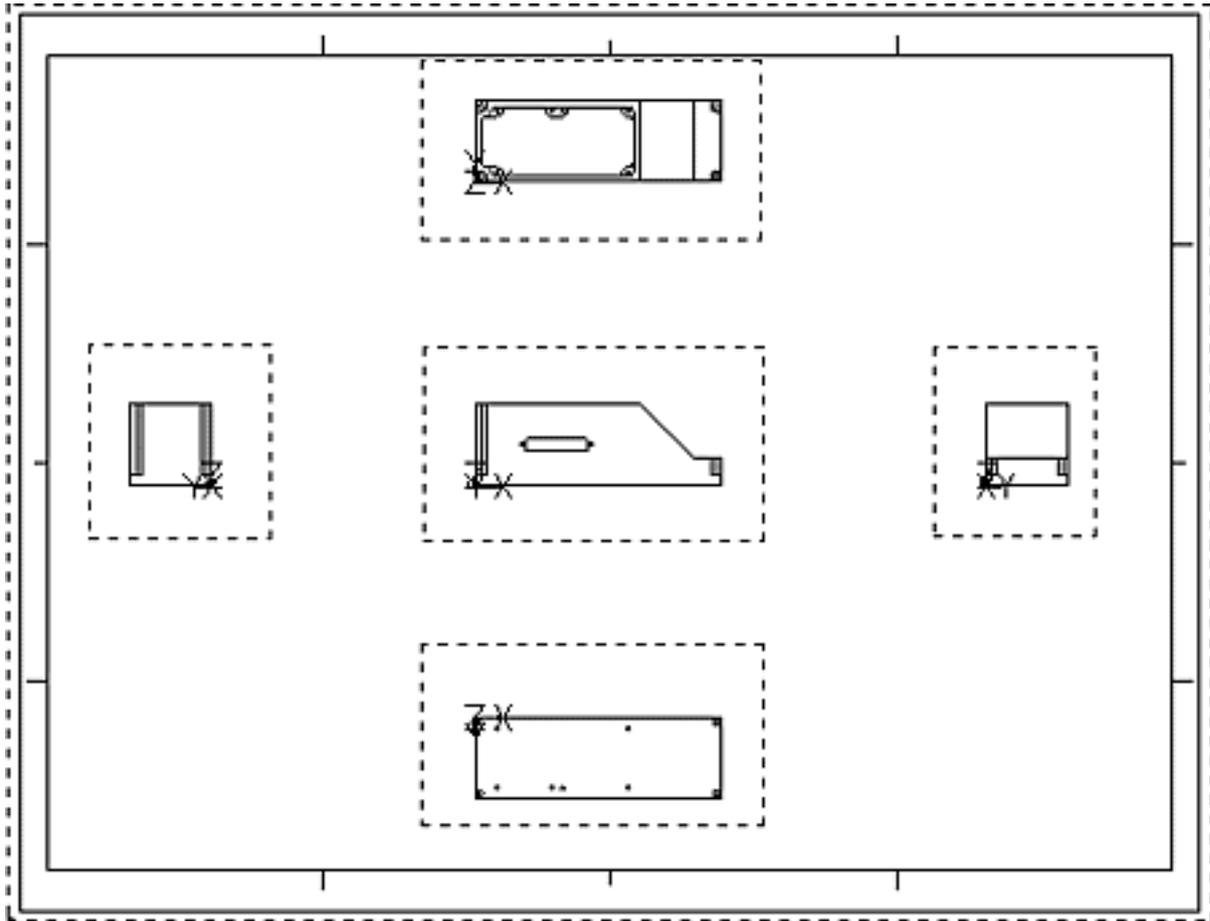
2. Use DEFINE VIEW FOLD to set up the other orthographic views. The newly created views are automatically aligned to the reference view.

Use DEFINE VIEW FOLD CLIP to automatically clip the view around the model, therefore reducing the number of graphical selections.

Use the **DEFINE VIEW FOLD ASSOCIATIVE** to associate the views with the reference views.

You can fold all the other views of the reference view as shown.

**Figure 2-2 Fold All the Views In the Reference View**



3. Add the detail views.

Several techniques to change the layout of the drawing and move views on the drawing follow.

## Aligning Two Views

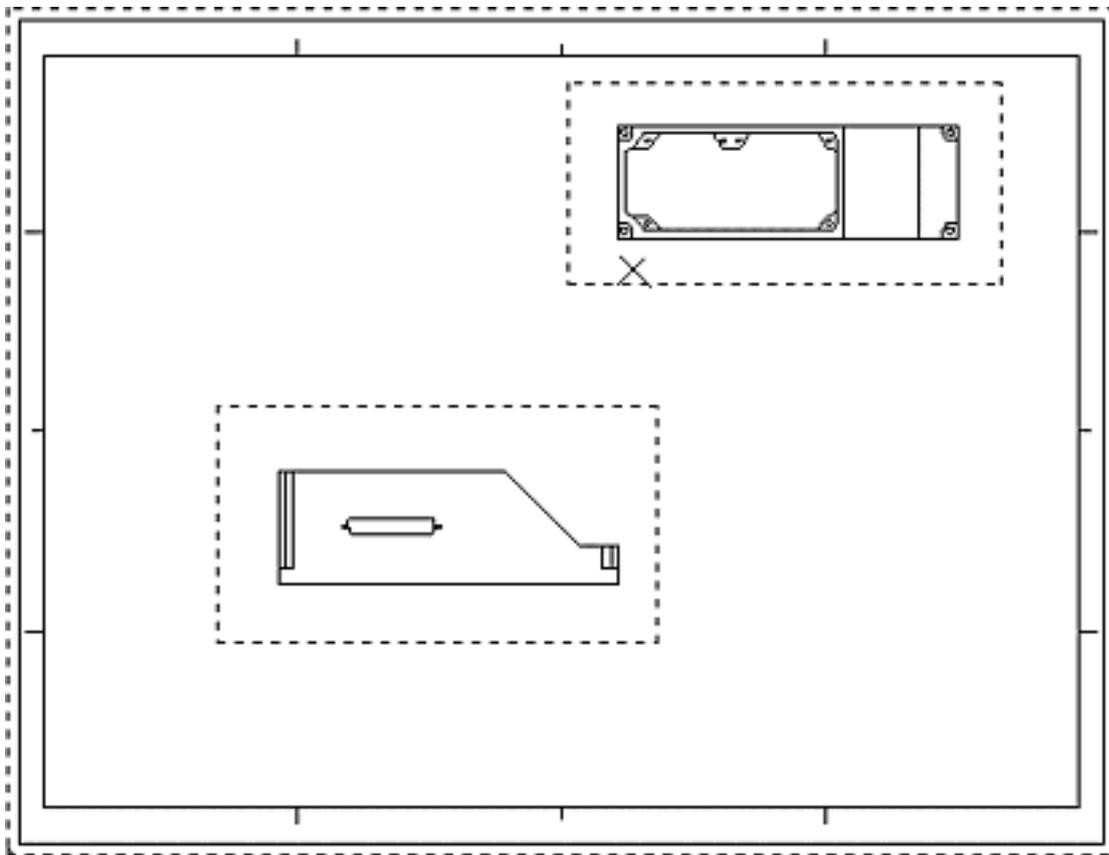
Use the following method to align two existing nonaligned views:

1. Choose the **Change Location** option from the **View** menu on the top bar. This option enables the **REVISE VIEW LOC** command on the command line.
2. Select the views that you want to align.

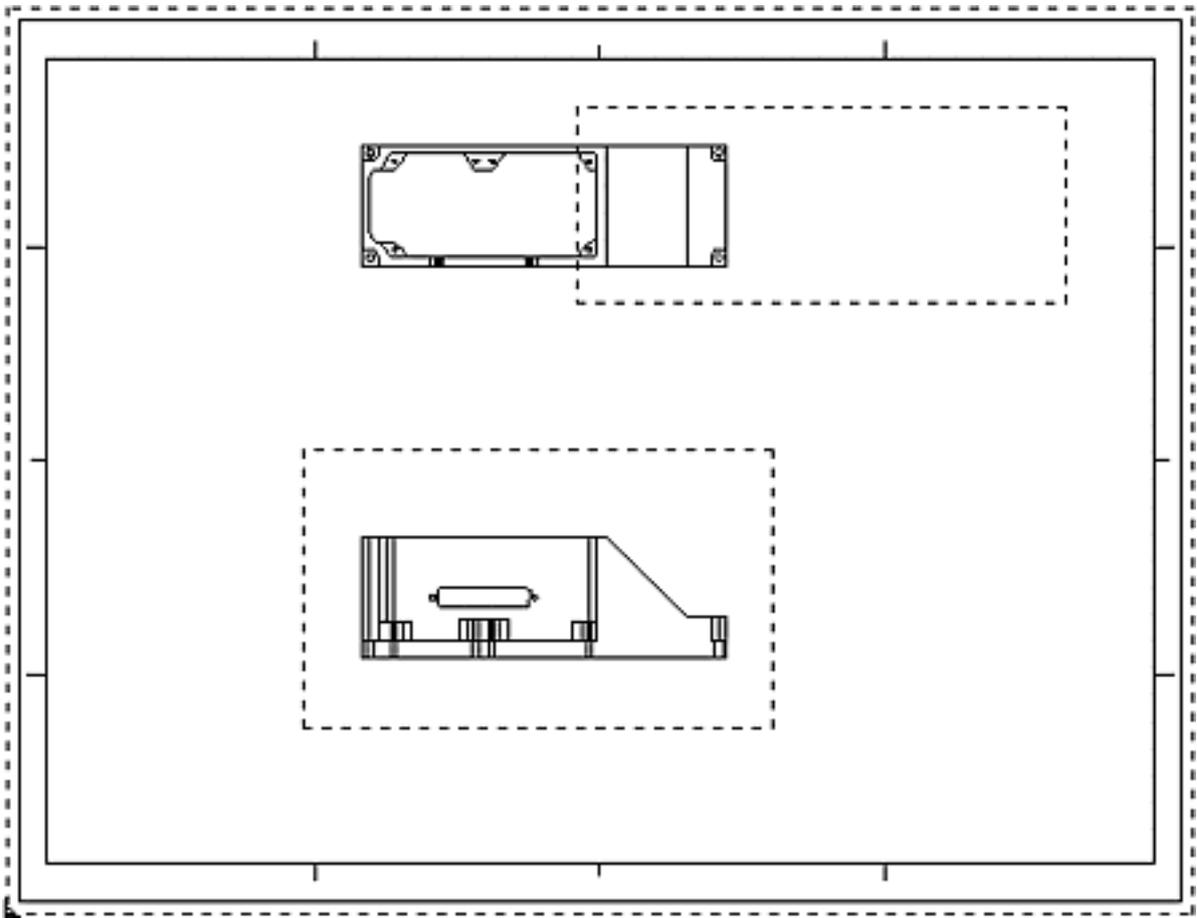
3. Select a location as a reference point on the model within the same view. (Pick a point that you can easily select in the other view.)
4. Select the proper direction using either the LOC X or LOC Y option. Using these options relocates the view on the x-axis or the y-axis of the drawing, depending on which of these axes is almost parallel to the path of the selected locations. The distance of the path is the distance between the first selected location and the second selected location.

Example for Aligning Two Existing Nonaligned Views: The following figures illustrate the procedure to align two existing views.

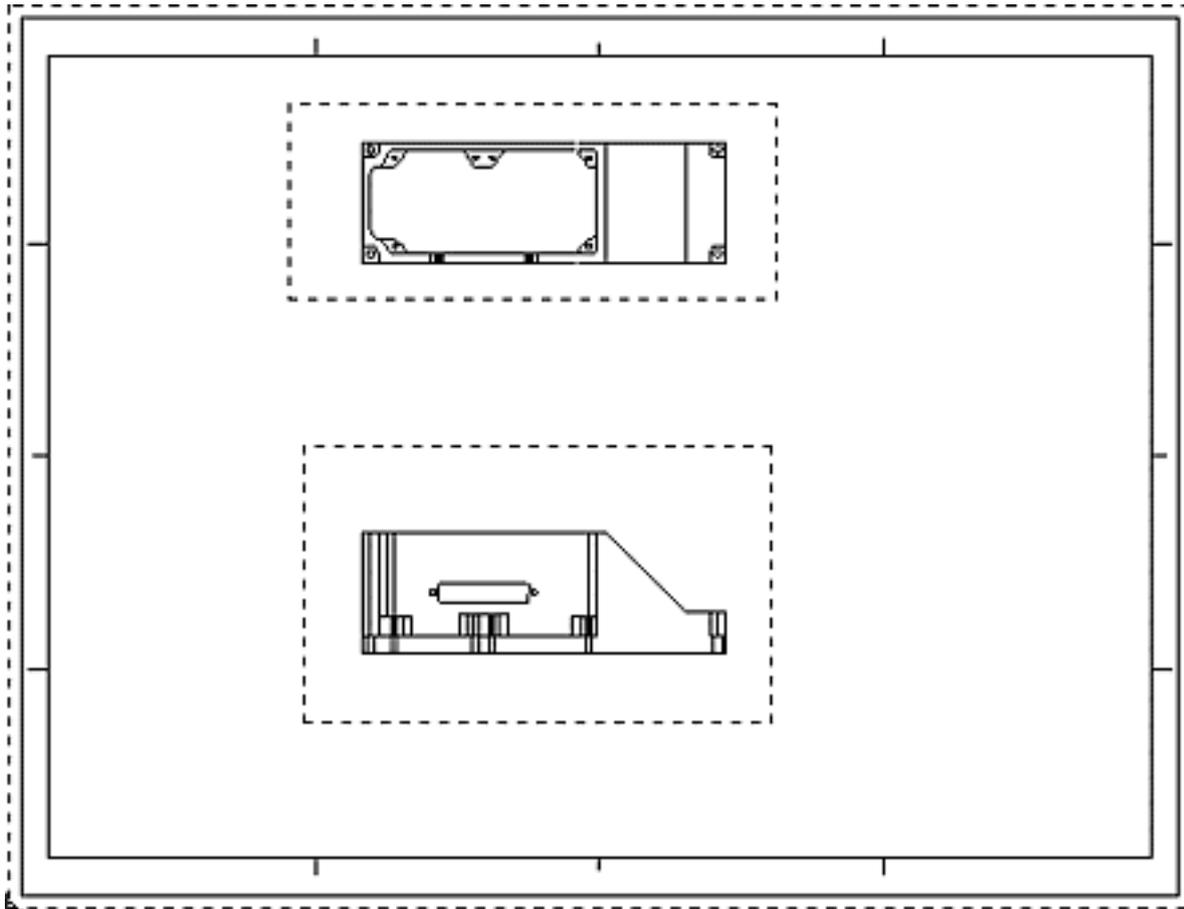
**Figure 2-3** Graphically Select the View and a Reference Point on the Model within the Same View



**Figure 2-4** Select LOC X and Graphically Select the Same Reference Point on the Model in the Other View



**Figure 2-5 The Two Views Are Now Aligned**



## Moving Views to Change the Layout

You can move views on the drawing to change the layout of the drawing. The following sections describe the various methods to move views on the drawing.

### Moving a View on the Drawing

Tips for moving a single view on the drawing follow.

- To move a single view or an entire row or column of views, use **SCROLL VIEW**.

- To scroll all the views in the drawing use the following command:

```
SCROLL VIEW: view <select views> MODEL loc <specify two locations>
```

- To move a specific view by name use the following command:

```
SCROLL VIEW: view NAME <specify view name> MODEL loc <specify two locations>
```

- To change the size of the view after moving it, use the **REVISE VIEW CLIP** command by choosing the Change View Size option from the **VIEW** menu.
- To move a single view on the drawing, use the **REVISE VIEW LOC** command by choosing the Change Location option from the **VIEW** menu.

Please note: **REVISE VIEW LOC** does not maintain the alignment between the different views if they are created without using the **FOLD** option.

- To move a view only in the x-direction or y-direction on the drawing, that is, either horizontally or vertically, use the following command:

```
REVISE VIEW LOC ALONG: view <dig view> location <dig two locations>
```

Select two locations that define a vector to move the view in the x-direction or the y-direction on the drawing. The view is moved along the larger value of the x-component or the y-component.

## Moving All Views on the Drawing

You can move all the views on the drawing. For example, if you want to create space on one side of the drawing, move all the views on the drawing using **REVISE VIEW LOC**. Select all the views graphically and run the command.

## Creating Detail or Scrap Views

Use **COPY VIEW** to create a detail view by copying a view. The **COPY VIEW** command allows you to define the detail view scale and area.

Please note: Detail views are also referred to as scrap views.

Release 10 Onward: **COPY VIEW** allows you to copy the HLR and section views in the part as well as in the CAMU environment

Revision 9 Onward: Use **COPY VIEW** to copy the HLR and section views in the part environment.

# Creating Section Views

Use DEFINE SECTION to create section views on a drawing. The other methods and commands that you can use are listed and described in the *Sectioning User Guide*.

## Using the DEFINE SECTION Command

The advantages of using DEFINE SECTION follow. The figures in this section show the general uses and the capabilities of the command.

### Advantages

The advantages of using DEFINE SECTION are as follows:

- Provides ease-of-use with no management overhead
- Creates the following:
  - Cutaway for HLRed section
  - Unfolded section
- Supports associative dimensioning
- Works on assemblies

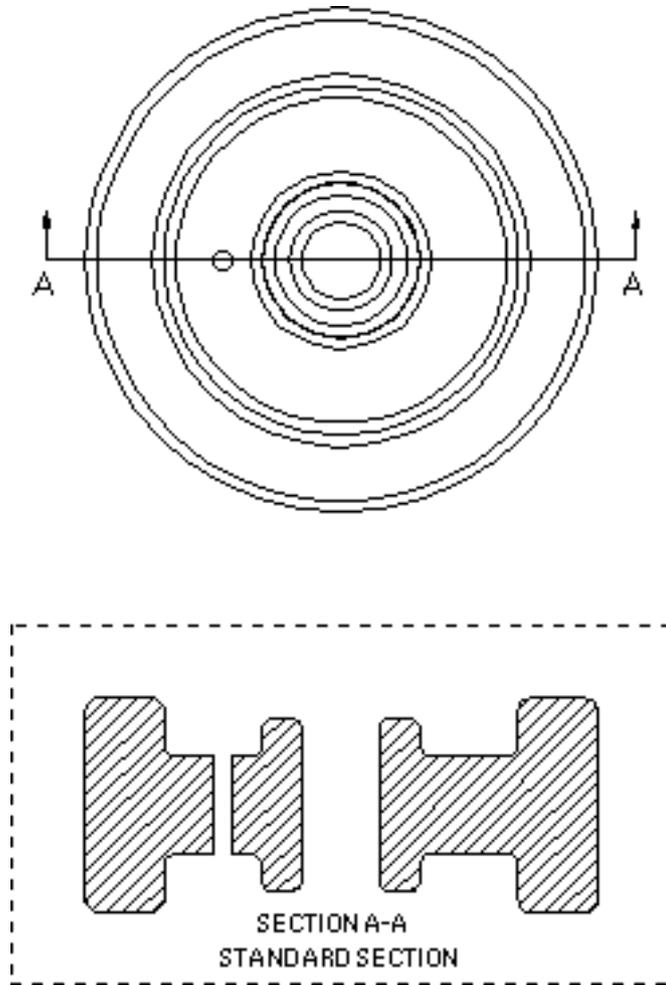
### Capabilities

The DEFINE SECTION command allows you to create the following items:

- The section profile only
- The sectioned solid representation with HLR performed on it
- An unfolded section

The following figures illustrate the sectioning capabilities.

**Figure 2-6 Standard Section**



**Figure 2-7 Cutaway Section with HLR**

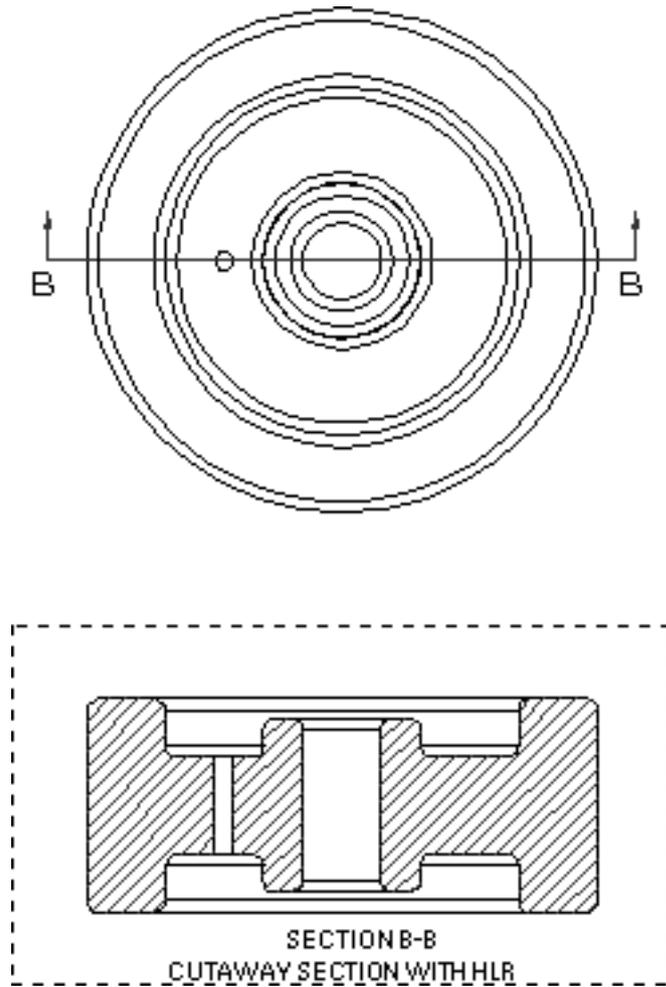
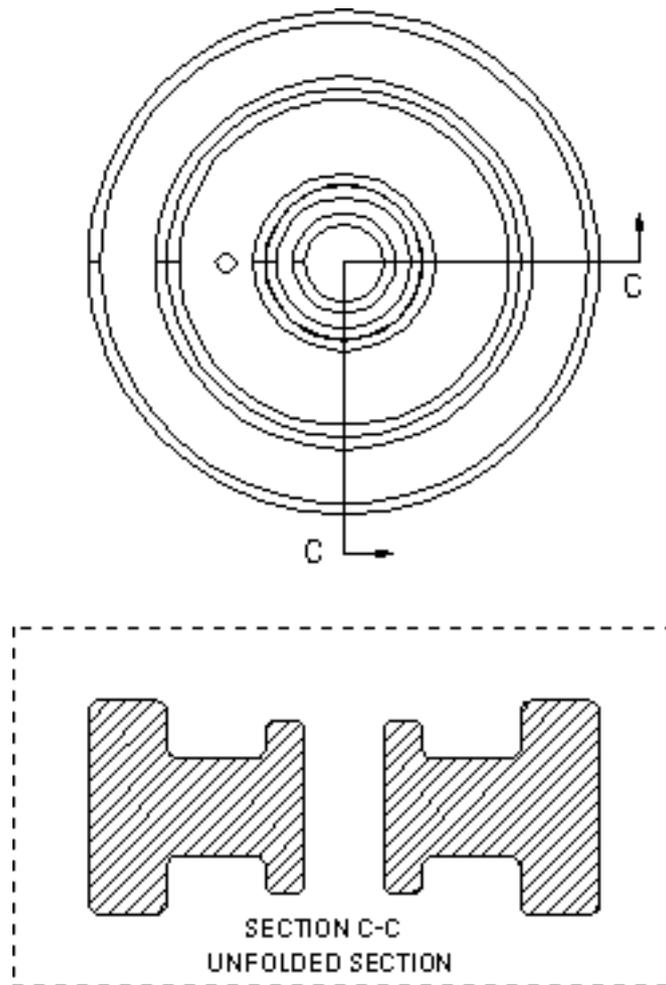


Figure 2-8 Unfolded Section



#### General Usage of the Sectioning Command:

```
Define section [Sectname <Section view name>]: view <select  
reference> [optionally select new view location] model ent <select  
sectioning tool> model ent <select solid to be sectioned>
```

## Creating a Section in a Separate View

You can create a simple, a cutaway, or an unfolded section in a separate view. Use a line or a string as a sectioning tool to create a separate section view. The orientation of the new view is determined by the first segment of the sectioning tool, which is the string. By default, the sectioning operation creates a view that fits tightly around the sectioning results.

Please note: You can also use a solid as a sectioning tool.

Release 13 Onward: You can also use a plane as a sectioner.

## Creating a Simple Section View

You can create a simple section view using a string as a sectioner by using the following command:

```
DEFINE SECTION SECTNAME versect
```

Please note: You cannot create a section view with the same name as that of an existing view or Cplane.

## Viewing Sectioning Results for Revision 8 and 9

Please note: Release 10 onward you do not need to use ECHO LAYER because unwanted entities are automatically cleaned up.

For Revision 8 and 9 you can view only the sectioning results in the Section View by following these steps:

1. Create the section as follows:

```
DEFINE SECTION SECTNAME <sectname> LAYER <layerno>:
```

2. Use the command

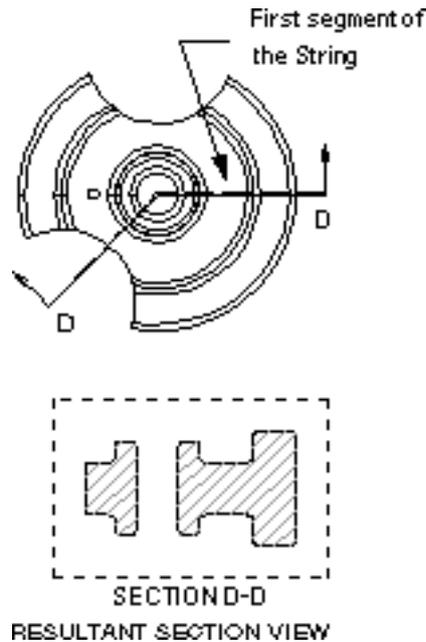
```
ECHO LAYER <layerno>: view (select the section view)
```

3. If some unwanted entities are still visible in the section view, then use ERASE ENTITY to remove such entities from the section view.

## Creating an Unfolded Section

To create an unfolded section, you must create a separate section view using the DEFINE SECTION SECTNAME <sectname> command. You cannot create an unfolded section in an existing view. The following figure illustrates the usage of UNFOLDED SECTION.

Figure 2-9 Creating an Unfolded Section Using a String as the Sectioning Tool



## Creating a Section in an Existing View

To create a section in an existing view, you must create a sectioning tool that is a solid of any shape. For example, you can create a stepped solid to create a stepped section. The **DEFINE SECTION** command subtracts the sectioning tool from the solid that is to be sectioned. You can construct the sectioning tool using various methods. For best results, use the following method for constructing a sectioning tool:

1. Use the explicit Wireframe option to create the section line. This geometry is used to display the section line on the drawing.
2. Add more wireframe to the original section line geometry to create a closed loop.
3. Sweep the closed loop into a solid.
4. Place all the geometries of the sectioning tool on a separate layer. Name the layer according to the section view. For example, if the section has a name A-A, name the layer as `TOOL_A_A`.

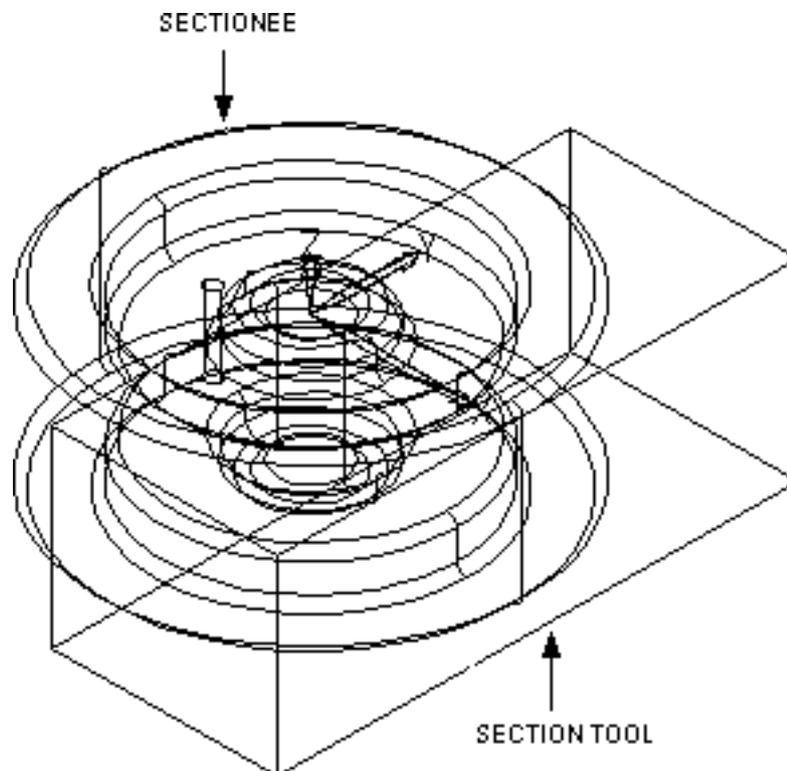
Release 13 Onward: You can also use a plane as a sectioner. When sectioning, create the sectioning tool in the Explicit environment. This ensures that CADDSS does not allow you to delete the sectioning tool until all the section views using the tool are first undefined. If you have created a parametric sectioning tool, ensure that you do not delete it before all the section views using the tool are undefined. The problems that can arise by deleting the sectioning tool are discussed in the section “Points to Note When Creating Sections” on page 2-25.

You can create cutaway sections in the existing views.

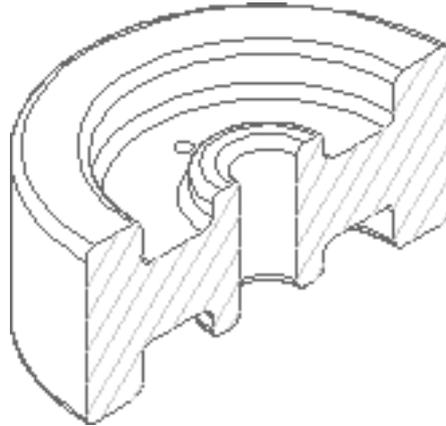
Please note: From Release 10 onward, all the options, except the INTERFERENCE and INTERSECT options that are available with the HIDE OBJECT command are also available with the DEFINE SECTION CUTAWAY command.

The following figures illustrate the creation of a section in an existing view.

**Figure 2-10 Sectionee and the Sectioning Tool Used by the DEFINE SECTION Command**



**Figure 2-11 Result of the DEFINE SECTION Command**



## Points to Note When Creating Sections

When creating sections, note the following points:

- Do not delete the sectioner that is part of the sectioning tool. Deleting the sectioning tool can corrupt the section, and, at times, you may not be able to undefine the section view using UNDEFINE SECTION. Use BLANK VIEW to suppress such a section view from the drawing. However, BLANK VIEW does not clean up the database with respect to the section view. Release 13 onward, if the sectioner is deleted, you can use REGENERATE SECTION to select a new sectioner. Select only the local entity as a sectioner. The section view is regenerated with the new sectioner.
- Avoid making any modeling changes to the sectioning tool because that can corrupt the section, and make it impossible for you to undefine the section. The best method to make modeling changes to the sectioning tool is to first clean all the sections that have this sectioning tool, using UNDEFINE SECTION. After making modeling changes to the sectioning tool, redefine the sections using DEFINE SECTION.
- Do not use HIDE OBJECT on a section view. If you do, even when the section view is cleaned up using UNDEFINE SECTION, all the garbage entities related to the HLR operation remain in the database.  
Use DEFINE SECTION CUTAWAY to create the HLRed section view.
- Do not use DELETE VIEW to clean up a section view. Use UNDEFINE SECTION to clean up the section view properly.

- Use REECHO ENTITY before updating or undefining the section views if you have used ERASE ENTITY to clean up some of the entities in the section view.
- Do not delete the section definition view. If the definition view is deleted, the REGENERATE SECTION command fails, and the command clears up the entire view. You can redefine the section again using DEFINE SECTION and another definition view.

Please note: If you define the section using DEFINE SECTION ALL, and then regenerate the section using REGENERATE SECTION, only those solids that are used during the DEFINE SECTION operation are regenerated. REGENERATE SECTION does not perform a getdata operation internally to select all the solids visible in the definition view.

Revision 9 Onward:

- When regenerating a section, you can select all the visible objects in a view by using REGENERATE SECTION ALLOBJECTS.
- To copy a section view, use COPY VIEW APPEARANCE. A copy of the section view is created without performing the sectioning operation.

Revision 8 Specific:

- If you delete the definition view, the section can be corrupted. It may not be possible to delete the section view.
- When you create multiple cutaway section views on the drawing, REGENERATE SECTION can fail. When there are multiple cutaway sections on a drawing, you must first undefine all the cutaway sections on the drawing and then define them again using DEFINE SECTION. All the other section views are then regenerated properly. If a drawing has a single cutaway section, then regeneration does not cause any problem.
- The COPY VIEW command copies only the view and not the sections in it.

## Updating a Section View

Use REGENERATE SECTION to update a section view. REGENERATE SECTION works correctly if the changes made to the model do not alter the topology of the model. For example, if you make a parameter change, then the topology of the model does not change and REGENERATE SECTION produces the required results. Also, when there are modeling changes such that the original workpiece is not lost, then REGENERATE SECTION produces the required results.

If the modeling changes are such that the original workpiece is lost, then **REGENERATE SECTION** does not generate results for that workpiece and cleans up the existing results related to that workpiece. Specific instances when the workpiece can get lost are when you perform the following actions:

- Remove the model in the Parametric environment using **DELETE ENTITY**.
- Create sections for a solid  $s_2$  and then perform a Boolean union between solid  $s_1$  and  $s_2$  where you select  $s_1$  first. In this case, the workpiece  $s_2$  is lost and you cannot regenerate the sections of solid  $s_2$ .

When you cannot regenerate the sections, the dimensions associated with these sections are lost.

**Please note:** When the regeneration of a section does not produce the required results, you must redefine the section using **DEFINE SECTION** and recreate all the dimensions.

To make modeling changes to the sectioning tool, you must remove all the sections using this sectioning tool before making the changes. For details, see “Points to Note When Creating Sections” on page 2-25. You can remove a section using **UNDEFINE SECTION**. Recreate the section using **DEFINE SECTION**.

## Rules for Updating a Section View

Observe the following rules when updating a section view:

**Revision 9 Onward:** If **REGENERATE SECTION** is unable to produce the required results, it cleans up the section view. If the sectioning tool is not present, a message is displayed asking you to clean up the section using **UNDEFINE SECTION**.

**Release 13 Onward:** If the sectioner is deleted, use **REGENERATE SECTION** to select a new sectioner. Select only the local entity as a sectioner. The section view is regenerated with the new sectioner.

**Revision 8 Specific:** The following rules are specific to Revision 8:

- If you have cutaway sections on the drawing, first undefine them using **UNDEFINE SECTION** and then redefine them using **DEFINE SECTION**.
- If you have deleted the sectioning tool, use **UNDEFINE SECTION** to clean up the sections.
- If the sectioning tool has been deleted, then **REGENERATE SECTION** may not clean up the section view and the database properly. It may therefore not be possible to delete the section view.

# Creating Centerlines

You can create centerlines for parts by using the conventional method, the parametric method, or by defining features.

## Using the Conventional and Parametric Methods

The conventional method of creating centerlines is more time-efficient than the parametric method. Because centerlines are associative, the conventional method reduces the work involved. Therefore, the conventional method is the preferred method for creating centerlines.

### Conventional Method

The conventional method of creating centerlines uses the standard set of commands provided by the Explicit environment. By default, centerlines are associative to the model geometry. If the model changes, centerlines are regenerated automatically and continue to be associative.

The general process for creating explicit centerlines is as follows:

1. Enter the Explicit environment.
2. Select the DRAFTING option from the task set menu to display the DRAFTING task set.
3. Use the INSERT CENTERLINES option to create the centerlines.

### Parametric Method

The parametric method of creating centerlines uses the parametric wireframe capabilities to define the wireframe that represents centerlines. This method allows you to create associative centerlines. Considerable work is involved in creating the initial centerlines on the drawing.

The general process for creating parametric centerlines is as follows:

1. Enter the Parametric environment.
2. Select the correct layer to store centerlines.
3. Create a parametric wireframe.
4. Enter the Explicit environment.
5. Change the font characteristics of the wireframe.

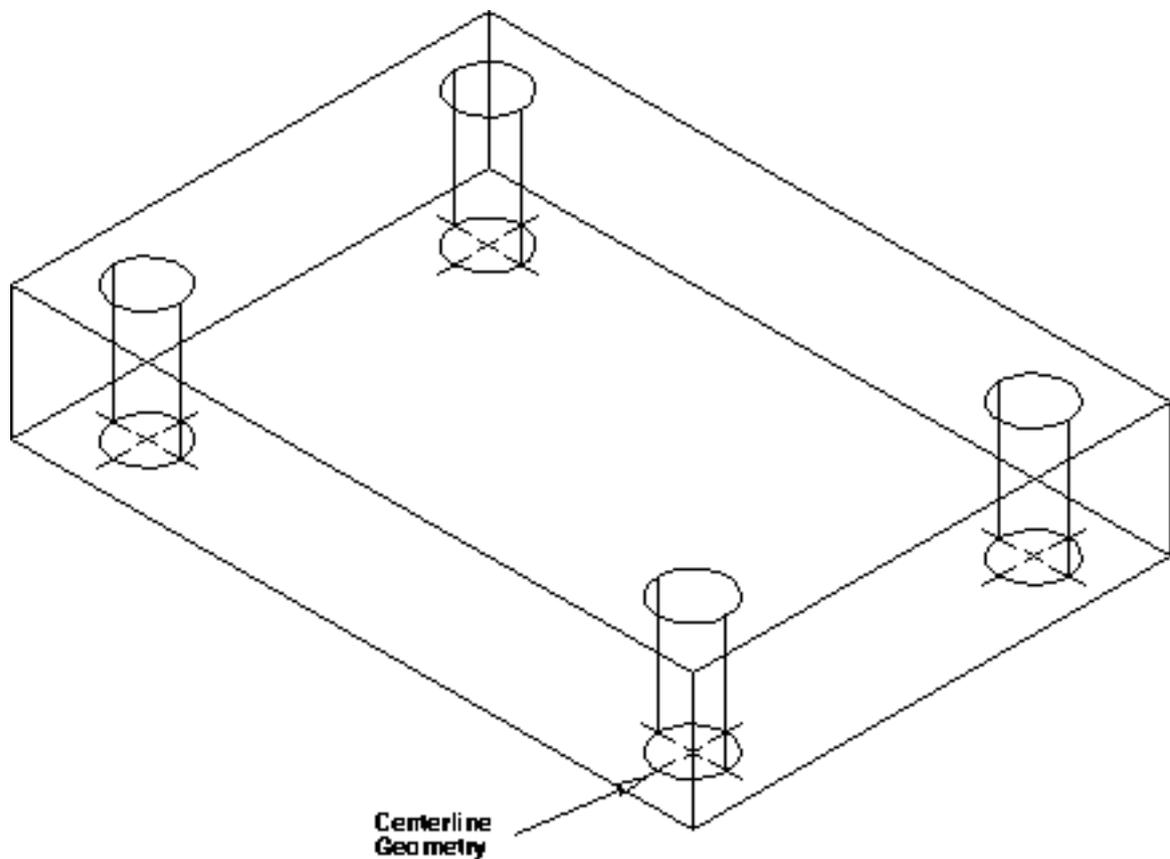
## Defining Features to Create Centerlines

Create a library of centerline feature commands to make the process of parametric centerline creation more user friendly. You can then insert a whole set of centerlines in a single operation. In the Parametric environment, use the Measure Length Diameter and Measure Length Points options available from the Measure menu on the top bar to reference the existing parametric geometry. This referenced parametric geometry can be used as input to define the size of the centerline geometry.

### Basic User-Defined Features

The following figure provides an example where a user-defined feature is used to conveniently insert the centerline geometry.

**Figure 2-12 Parametric Centerlines Are Associative to the Parametric Model**



You can insert the feature shown in the figure with a customized option using the following commands:

```
select layer <center line layer number>
```

```
select library centerlinelib
```

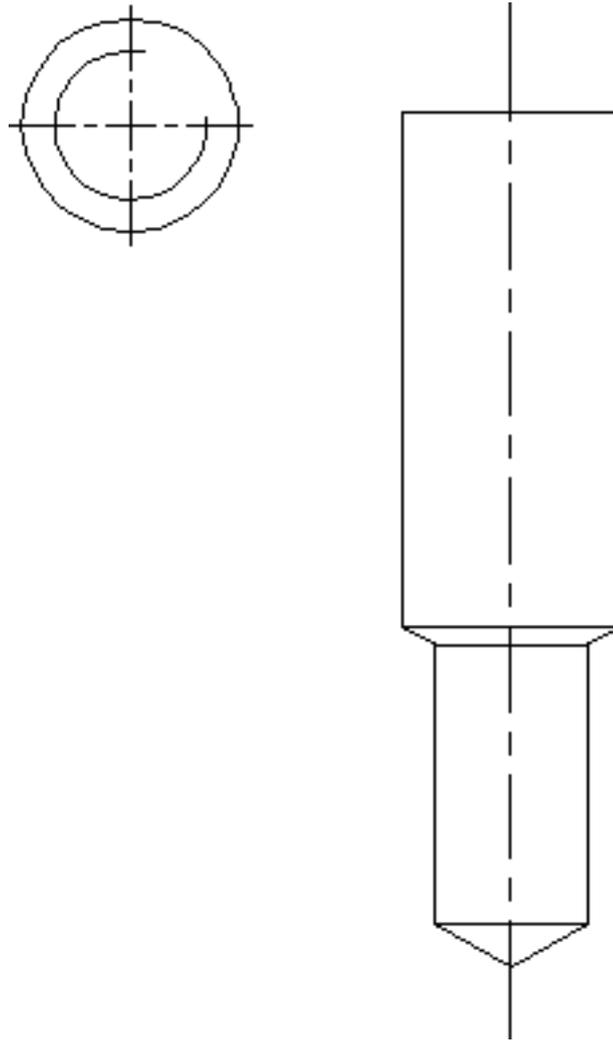
```
insert feature A_centre_line_hole_top diameter measure length  
diameter <dig diameter of existing hole> origin <select origin of  
existing hole> go
```

## Advanced User-Defined Features

You can build intelligent features when working with the process of parametric centerline creation. For example, you can build a feature that contains the solid geometry for a tapped hole, including all the detailing geometry needed to complete the drawing. You can define the critical dimensions by defining an `if then else` list with constraints. For example, `if diameter = 10 then inside_diameter = 8.8`. This entire feature can then be inserted by specifying a single value for the diameter of the hole, that is, 10 for M10.

The following figure illustrates a more advanced user-defined feature that contains the solid geometry and the wireframe geometry for the detail drawing of a tapped hole. You can insert the entire geometry set on the parametric model as one operation.

**Figure 2-13 Advanced User-Defined Feature**



## Creating HLR Views

Use the following command to create HLR views:

```
HIDE OBJECT [ALL] [other modifiers]: view <select view> [model ent  
<select object to be HLRed>]
```

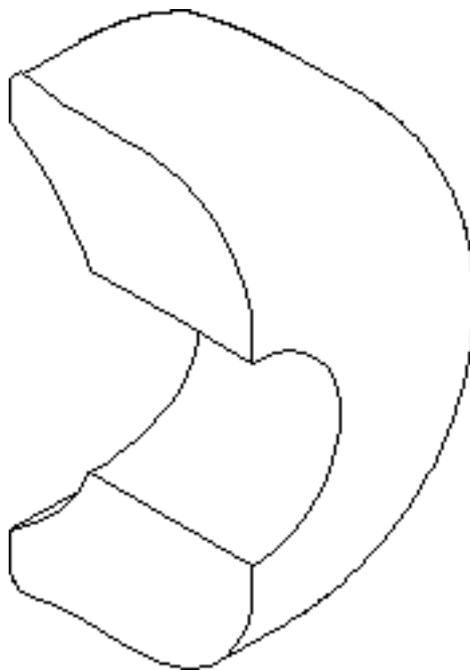
Please note: Use **COPY VIEW APPEARANCE** to copy the HLR view. This command copies the HLR view without performing the HLR operation. This saves the cost in both time and effort involved in performing HLR.

Use **HIDE OBJECT ALL** to perform HLR on the entire view. Otherwise, HLR is performed only on the entities that you select. Use **HIDE OBJECT TAN** to blank all the tangent edges of a solid in an HLR picture. This option works on all solids created with a rotational sweep, a linear sweep, or fillets.

### Example of an HLR View

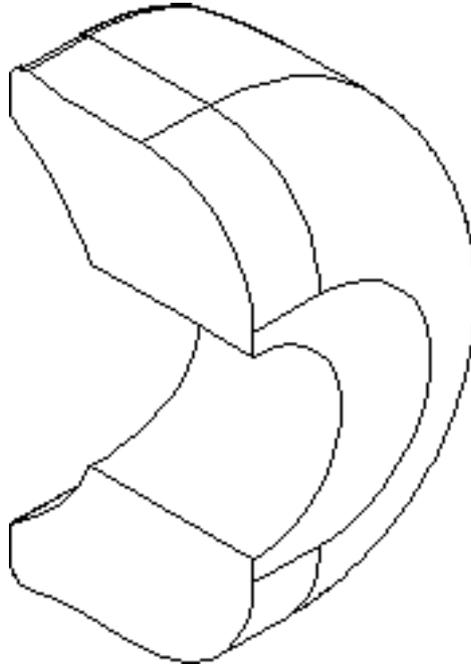
The following figure illustrates the usage of **HIDE OBJECT TAN**.

**Figure 2-14 Hide Object with the TAN Option**



The following figure illustrates the usage of HIDE OBJECT without the TAN option.

**Figure 2-15 Standard Hide Object without the TAN Option**



Use HIDE OBJECT DASH if you want the hidden lines to appear dashed. If you do not use HIDE OBJECT DASH, the hidden lines are blanked.

For details about other options available with HIDE OBJECT, see *Hidden Line Removal (and AEC HLR) User Guide and Menu Reference*.

## Ensuring Quality of the HLR Image

You can create the HLR image either by using the analytical engine or the faceted engine. For high quality HLR images, observe the suggested best practices for using the faceted or analytical engine.

For achieving good results when creating HLR images ensure the following:

- When identical curves overlap, HLR cannot produce good results. For example, during the designing process, you can create many construction curves from which the final solid is created. If these construction curves remain in the view, they overlap the solid generated from them. Therefore, HLR cannot produce good results when HIDE OBJECT is applied to the view. Remove the construction wireframe from the view either by deleting the curves or by moving them to a layer that is not echoed in the view upon which HLR will be performed. After moving this unwanted wireframe to a layer that is invisible in the view, you can use HIDE OBJECT in the view.

Please note: It may not be easy to detect whether an unwanted wireframe is visible in the view. Use a combination of the SOL, SURF, and AEC options to eliminate the wireframe.

- You must remove duplicate solids residing at the same position. Otherwise, you may not achieve the required results.

## Using the Faceted Engine

Please note: The faceted engine is available only from Release 11 onward.

The faceted engine provides better performance for a model with more curved objects than planar objects. The faceted engine provides you with an easy-to-use slide bar to control quality and performance. This bar is called the quality-versus-performance bar. Use HIDE OBJECT FACETED to enable the faceted engine to perform HLR. By default, the slider is at the center of the bar.

If the HLR results are not satisfactory, then update the image with a better quality factor. Move the cursor to the right and perform the HLR operation again. Continue moving the cursor to the right and performing HLR until you have the desired quality. Moving the cursor to the left results in faster HLR performance and an image that is less accurate.

Revision 12 Onward: The faceted engine is also available in the CAMU environment for performing HLR operations. Using the faceted engine in the CAMU environment results in good performance.

## Using the Analytical Engine

To obtain high quality HLR images using the analytical engine, observe the following best practices:

- To achieve good quality in less time, use a combination of quality control options at each step. If that combination does not give good results, unhide the view and try the following suggestions:
  - First use HIDE OBJECT without any quality control options such as REFINE, TOL, LIMBTOL, SEG, INTERFERENCE, and INTERSECT.
  - If the quality of the resultant HLR image is not good and the model has interfering objects, use the INTERFERENCE option.
- If after using the INTERFERENCE option the quality of the resultant HLR image is still not good, and the model has a lot of curved objects, use HIDE OBJECT REFINE to get better results.

Please note: If you use HIDE OBJECT REFINE, you cannot use the TOL, LIMBTOL, SEG, and INTERFERENCE options.

- If the problem still persists for the limbs, try varying the limb tolerance using the LIMBTOL option.
- If the resulting quality is still not good, try varying values of the TOL, LIMBTOL, and SEGMENT options.

The tolerance that is used internally by HIDE OBJECT is listed at the end of the completion of the HLR process. The results of HIDE OBJECT is as follows:

```
##
## HIDE OBJECT Dash DLen 4 SLen 3: view ddd; MODEL ent d
Processing Parts
Generating Limbs and analyzing faces...
Generating Limbs and analyzing faces...
Generating Limbs and analyzing faces...
Tolerance used for HLR = 0.26874
##
```

## Updating an HLR View

Use UPDATE HLRIMAGE to update the HLR view if the model has undergone changes. When UPDATE HLRIMAGE is applied to an HLR view, the previous results are cleaned up and the new results are computed and displayed in the view.

## Updating an HLR View from Release 11 Onward

The following points save you cost and time when updating the HLR views from Release 11 onward.

- The new, easy-to-use user interface provides you more control when updating an image.
- The Merge option enables you to have local control over the HLR image. The following examples indicate how UPDATE HLRIMAGE MERGE saves time.
  - If you have used the faceted engine with the default quality factor and the picture is bad only in a very small area, use UPDATE HLRIMAGE MERGE and select only the region with bad results. This action saves time in achieving an overall good image.
  - After performing HLR on the explicit views, the model in the Parametric or the Explicit mode can change because of the addition of new solid entities. In this case, update the views using UPDATE HLRIMAGE MERGE. This command allows you to select only the additional entities instead of selecting the entire view again for HLR, saving a lot of time.
- If you are interested in only blanking all the dashed lines quickly in the HLR view, use UPDATE HLRIMAGE BLANK. This saves you the cost of running the engine and saves you time.
- Like dashed lines, tangent and topological edges can be displayed or removed without running the engine. To do this, use UPDATE HLRIMAGE with the SHOWTANEDGES, SHOWTOPOEDGES, REMOVETANEDGES, and REMOVETOPEDGES options.
- If you are not satisfied with the results obtained from one engine, change to the other engine during the update process and save the efforts of selecting the objects or the options again.

## Unhiding the HLR View

To unhide the HLR view, use UNHIDE OBJECT. Do not use DELETE VIEW to clean up the HLR view.

Please note: If all the hidden lines do not reappear after using UNHIDE OBJECT, use REECHO ENTITY and select the solid in that particular view.

## Layering Conventions and HLR

Consider using a defined layering convention for your application needs. For example, you can define a separate layer on which you can place each of the following entity types:

- Finished solid
- Structural model
- Piping system
- Drawing form parts
- Dimensions
- Construction geometry
- Reference geometry

Analyze your needs to define the layering convention that best suits your application. Some examples of application needs are:

- Some AEC applications use specific layers
- Offsetting of layers is necessary when using CAMU

Getting the layering conventions right and making use of them benefits you at all stages of the design, modification, and drawing production process. After defining the layering convention and creating a part using this layering convention you can easily spot:

- Any finished solid or surface model (explicit or parametric) in the part
- The layer each application is on
- The layers that have been used for construction or application reference geometry

After creating the part using a defined layering convention, you can easily exclude the unwanted layers from the views before you perform Hidden Line Removal (HLR).

Reference and construction geometry, for example, surfaces, construction solids, and so on, affect the time and quality of any HLR operation to a large extent. This is because you are attempting to hide coincident objects against each other as well as the rest of the scene.

After understanding the layering convention, you can use it to include only those layers that are necessary in each view to create the required HLR drawing.

Reducing the number of visible layers in each view speeds up hidden line removal in any given view or combination of views.

As a best practice do either of the following:

- Use different layers for the reference geometry when creating AEC objects and exclude these layers when performing hidden line removal.
- Transfer all AEC objects on to the current visualization layer before performing any HLR operation, either faceted or analytical.

## Adding Dimensions

The best practices for dimensioning parts are described in the following sections. In Part mode, it works best to use the explicit dimensioning method. The two dimensioning methods that you can use are:

- Explicit dimensioning
- Design dimensioning in the Parametric environment

Please note: The CAMU environment does not allow you to create design dimensions. Therefore, you must use explicit dimensions in CAMU.

Design dimensions do not support the following dimension types:

- Parametric fillet
- Chamfer
- Offset

## Creating Explicit Dimensions

Explicit dimensioning provides you with easier methods to create dimensions on a drawing as compared with design dimensioning. Even when using design dimensions, a few explicit dimensions are needed to complete the drawing. By default, the dimensions created in the Explicit environment are associative. When you make modeling changes to the model, the dimensions continue to remain associative unless the entities to which the dimensions are associated undergo major topological changes. If a dimension does not remain associative after making modeling changes, then it is dissociated and marked. To reassociate the dimensions that were dissociated, use `CHANGE DIMENSION ASSOCIATIVE`.

Please note: The previous statement is true only if `SELECT DIMENSION RETAIN` is set to `ON`. If `SELECT DIMENSION RETAIN` is set to `OFF`, the dimensions that do not remain associative after the model has undergone topological changes are deleted.

## Associative Dimensions

Dimensions that are attached to section and HLR views are also associative. Dimensions applied prior or subsequent to `HIDE OBJECT` are retained even after the `HIDE OBJECT`, `UPDATE HLRIMAGE`, and `UNHIDE OBJECT` operations are performed on the view. Dimensions that are applied to the limbs are also retained after the `UPDATE HLRIMAGE` operation is performed. However, dimensions that are applied to the limbs are lost after using `UNHIDE OBJECT`.

Dimensions that are applied to the section curves are retained even after the section is regenerated using **REGENERATE SECTION** in the view.

Revision 8 specific : In CADD5 Revision 8, the dimensions that do not remain associative after the modeling changes are dissociated but are not marked. In order to see the dissociated dimensions, use **FLAG DIMENSIONS NONASSOCIATIVE**. This command marks all the dimensions that are nonassociative, enabling you to see the dimensions that need to be checked.

Release 13 Onward: By default, annotations such as a balloon, label, finish symbol, weld symbol, datum symbol, datum target, text, and so on that are created in the Explicit environment, are associative. This associativity is preserved even after Parametric changes. If an annotation does not remain associative after modeling changes, then it is disassociated and marked.

You can insert, modify, delete, and verify annotations and text in the CSD environment. To insert different types of annotations, use the following hot keys:

CTRL + B	Balloon
CTRL + T	Text
CTRL + L	Label
CTRL + F	Finishing Symbol
CTRL + W	Weld Symbol
CTRL + S	Datum Target
CTRL + D	Datum Symbol

You can associate or disassociate any annotation using the **CHANGE DIMENSION** command.

## Tips for Creating Dimensions

Please note: The creation of dimensions using Context Sensitive Dimensioning (CSD) is possible only from Revision 9 onward.

This section provides some tips that help you obtain best results during the dimensioning process.

- You can create and modify dimensions in the CSD environment. This environment provides an efficient and easy-to-use interface for creating dimensions. CSD reduces the time for making changes to dimensions. Operations such as applying tolerances, changing positions, and adding a prefix or suffix, and other operations that are supported by CHANGE DIMENSION, are done much faster.

Release 13 onward, you can drag Draw mode entities such as Tnodes, Cnodes, or Nfigs using CSD.

- By default, SELECT DIMENSION ASSOCIATIVE DTMVERT is ON from Release 10 onward. It creates vertex associative dimensions. Do not toggle SELECT DIMENSION ASSOCIATIVE DTMVERT to OFF unless required. Vertex associative dimensions continue to remain associative even after the model has undergone topological changes. If the topological changes are such that the vertex does get lost, then the vertex associative dimension is dissociated and marked.

## Using Design Dimensions — A Parametric Approach

The advantages and the disadvantages of using design dimensions follow.

### Advantages

The advantages of using design dimensions are as follows:

- You can achieve approximately 98% of dimension associativity after design changes.
- When the model undergoes major changes, regeneration of one of the user dimensions gives you an indication of which dimension to correct.

### Disadvantages

The disadvantages of using design dimensions are as follows:

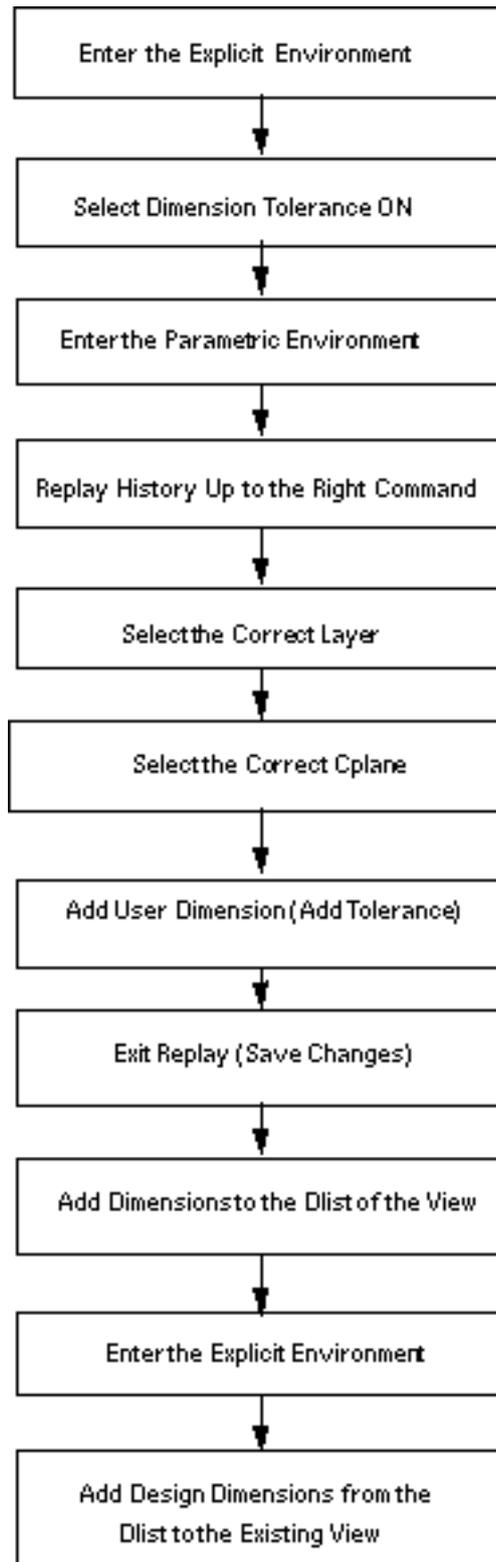
- Because design dimensions can only be defined in the Parametric environment, you have to switch between the Explicit and the Parametric environments to complete the drawing.
- Not all dimension types are currently supported using design dimensions. Therefore, you may still need to use the explicit dimensioning method.

- Tolerances on the design dimensions must be added in the Parametric environment.
- Design dimensions are not supported in the CAMU environment.

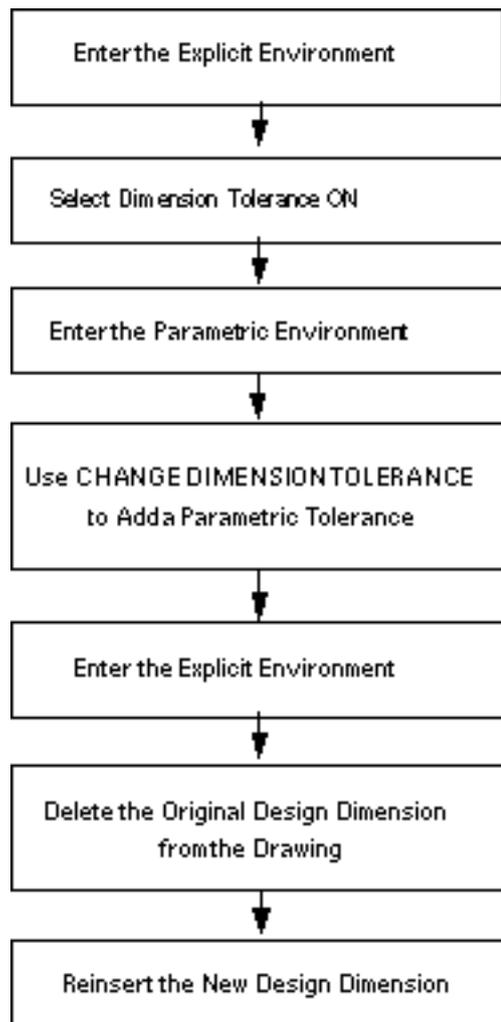
## Creating Design Dimensions

These figures show the process of adding a design dimension to an existing drawing or adding a tolerance to an existing design dimension.

Figure 2-16 The Process for Adding a Design Dimension to an Existing Drawing



**Figure 2-17 The Process for Adding a Tolerance to an Existing Design Dimension**



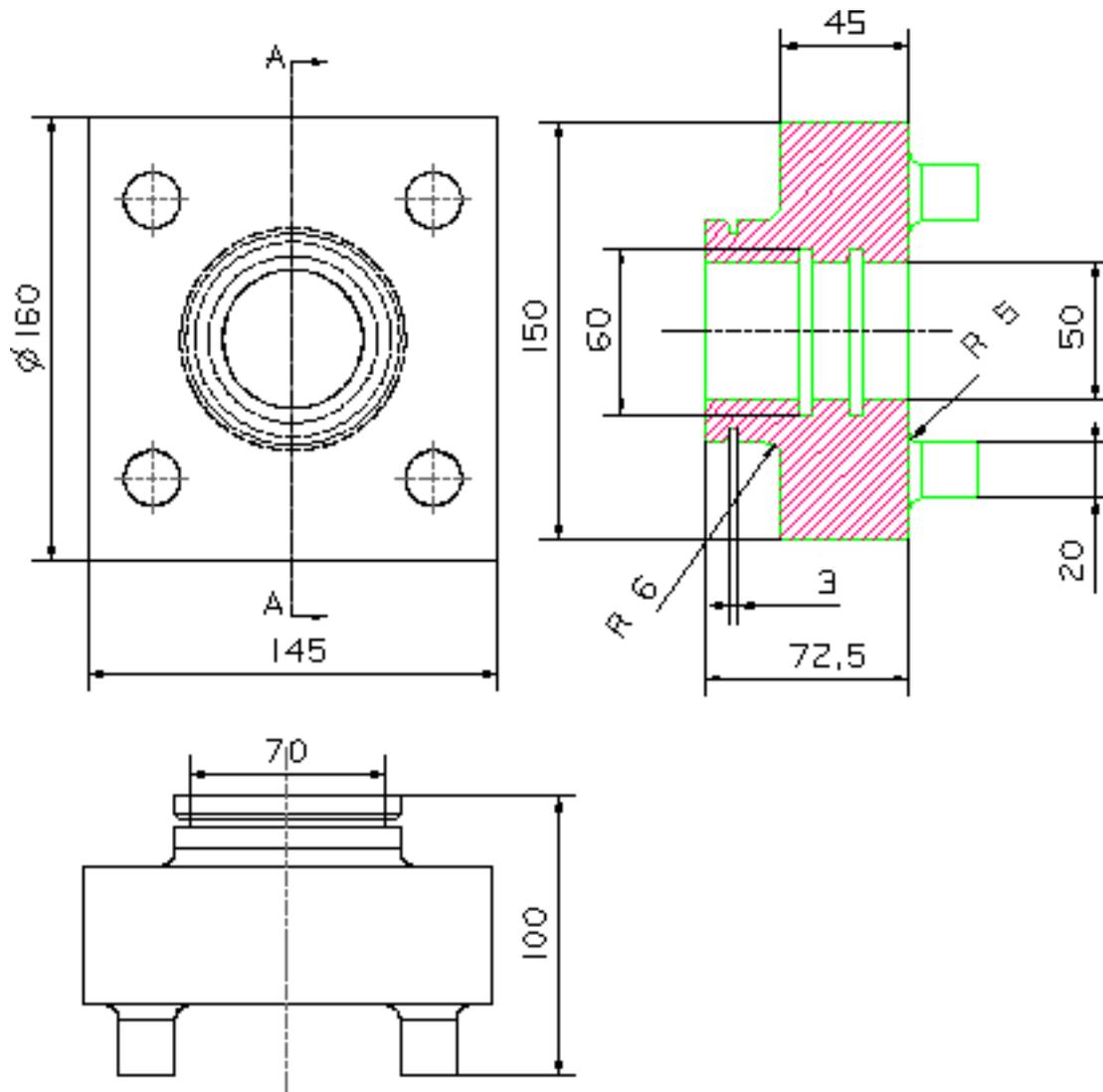
## Managing Design Dimensions

You must organize your work to be able to work with design dimensions while setting up the model. Some recommendations that can help you work with design dimensions follow:

- Insert design dimensions using a `dlist`. Define the `dlist` in the Parametric environment.
- Create one `dlist` for every explicit view and give the `dlist` the same name as the explicit view, for example, `dlist TOP, FRONT, SECTION_A_A`.
- Place the design dimensions for a single view on one layer and give the layer the same name as the explicit view.



Figure 2-19 Design Dimension As They Appear in the Explicit Drawing



## Solving Design Dimensioning Issues

Some of the known issues with design dimensions and their solutions follow.

- Sketcher parameters

Do not use Sketcher parameters as design dimensions. When reapplying the sketch after a change, the Sketcher parameters are recreated and are no longer a part of the `dlist`. These parameters, therefore, disappear from the drawing.

To avoid this, recreate the dimension scheme using user dimensions in the Parametric environment. These dimensions are associative to the sketch and remain on the drawing.

- Corrupt arrowheads

When you flip the arrowheads of a diameter design dimension to the outer side, the dimension on the drawing may become corrupt. This happens for very small diameter values. To avoid this:

- Delete the corrupt design dimension using `DELETE ENTITY`.
- Reinsert the diameter design dimension.
- Adjust the dimension arrowhead length as required. This adjustment corrects the arrowhead size and results in a valid dimension on the drawing.

- Cplane orientation

You must create the design dimensions on a Cplane that reflects the orientation of the view of the explicit drawing in the Parametric environment. The x-direction and y-direction of the Cplane must match the x-orientation and y-orientation of the view. When this Cplane is active, it shows up with a horizontal x and a vertical y in the view that is going to be used for the design dimensions.

You may need to create extra Cplanes. Create a Cplane in the Parametric environment and then select the same Cplane while inserting the design dimensions in the Explicit environment.

## Filing the Part for Saving the Drawing

Before filing a part you must ensure that you have:

- Saved the setup for the drawings
- Established a valid database

To prepare your part for filing perform the following steps:

1. Use `SAVE SETUP` to save all the drawing configuration information in the Explicit environment. The following section describes this command in detail.
2. Use `CHECK DBASE`. For details on validating a part, see “Checking the Validity of Components” on page 2-8.
3. File the part in the Explicit environment.

Please note: Release 13 onward, you can file a part in the Parametric environment. The Explicit database and the most recently active Explicit drawing are automatically updated. If no Explicit drawing is active at the time of filing the part, a new drawing with the name `default` is created and updated.

### Using the `SAVE SETUP` Command

Use `SAVE SETUP` to save the drawing after the different views have been created. The `SAVE SETUP` command is used to save the view clipping, zoom, orientation, scroll, origin, and visible layers for each view in the current drawing. You can create multiple `SAVE SETUP` files to keep several drawing configurations. This information is stored in a text file and in the following path.

```
/partname/_bcd/setup<drawingname-name>
```

You can enter the name when you use `SAVE SETUP`. By default, the name is `setup`. The syntax of the `SAVE SETUP` command is:

```
SAVE SETUP <name of the setup file; When not specified, default is setup>
```

Please note: If you do not specify a file name for the setup file, then the file is named `setup` by default.

## Using the RESTORE SETUP Command

You can restore the view information by using **RESTORE SETUP**. This command restores the visible layer information if you accidentally performed **ECHO LAYER ALL**. The syntax for the **RESTORE SETUP** command is:

```
RESTORE SETUP <name of the setup file; When not specified, default  
is setup >
```

## Updating Detail Drawings for Parts

Update the drawings to synchronize them with the changed model. The update indication mechanism available from Release 10 onward helps you determine the views that require an update and the dimensions that require reassociation.

You can update the entire drawing by using the options available from the Drawing Data menu. To update the drawing:

1. Select the Verify (?) option from the top bar menu. The Verify(?) menu appears.
2. Select the Drawing option. The Drawing Data menu appears. You can update an entire drawing using the Drawing Data menu.
3. Select the Allviews option from the Drawing Data menu to display the list of all the views in the drawing.
4. Use the View(s) due for update option to display the views that require updates in the runtime list.

Double-click the highlighted view name to display the View Data options for more information on the view that needs updating.

## Updating a Drawing for Minor Modeling Changes

When changes to the model are minor, for example, only parametric changes are made to the model, perform the following steps to update the drawing:

1. Update all the section views. Use REGENERATE SECTION ALL to regenerate all the section views in the drawing with one command.

Please note: For Revision 8, first undefine all the cutaway sections on the drawing, and then use REGENERATE SECTION ALL to regenerate the drawing. Redefine the cutaway sections using DEFINE SECTION.

2. If regeneration fails for any section view, you can:
  - Redefine the section view using DEFINE SECTION.
  - Clean up the view using UNDEFINE SECTION.
3. Update all the HLR views.
4. Reassociate marked dimensions and annotations where necessary.
5. Add section views or HLR views and dimensions if required.

## Updating a Drawing for Major Modeling Changes

When the model undergoes major changes, for example, the topology of the model has changed completely and the original workpiece is lost, perform the following steps to update the drawing:

1. Apply **REGENERATE SECTION ALL**. The sections that you cannot regenerate because the sectionees have undergone major changes are cleaned up by **REGENERATE SECTION**.

Please note: For Revision 8, clean up the section views using **UNDEFINE SECTION ALL** and then redefine the section views using **DEFINE SECTION**.

Release 13 onward, if the sectioner is deleted, you can use **REGENERATE SECTION** to select a new sectioner. Select only the local entity as a sectioner. The section view is regenerated with the new sectioner.

2. Redefine the sections that were cleaned up by **REGENERATE SECTION** using **DEFINE SECTION**.
3. If regeneration fails for any reason for any HLR view, you can:
  - Update all the HLR views using **UPDATE HLRIMAGE**.
  - Use **UPDATE HLRIMAGE** for all the HLR views. If **UPDATE HLRIMAGE** fails, use **UNHIDE OBJECT ALL** for a particular view and recreate that HLR view using **HIDE OBJECT**.
4. Reassociate the marked dimensions, centerlines, or annotations where necessary.
5. Add section views or HLR views and dimensions or centerlines if required.



# Creating and Updating Assembly Detail Drawings

---

This chapter discusses the best practices for creating assembly detail drawings.

- Considerations for Assembly Drawing Creation
- Points to Note Before Activating an Assembly
- Creating Assembly Detail Drawings
- Setting Up CADDs
- Activating an Assembly
- Activating a Drawing in the Explicit Environment
- Defining Required Views
- Creating Section Views for Assembly Drawings
- Creating HLR Views on Assembly Drawings
- Validating an Assembly
- Adding Dimensions, Centerlines, or Annotations to Assembly Drawings
- Saving the Setup and Filing the Drawing
- Updating the Assembly Detail Drawing

## Considerations for Assembly Drawing Creation

Various considerations and recommendations are involved when deciding which procedure to use to create assembly detail drawings. Considerations are as follows:

- Consider the performance when creating assembly drawings when the complexity and size of the assembly increases.
- Synchronize the drawings with the changes to the components and the changes to the assembly.
- Remember that the complexity of the problem increases in a concurrent scenario where the assembly is shared across users and is constantly changing.

Considering the previous points, start creating an assembly drawing only after the design of the assembly has reached a certain logical stage of completeness.

# Points to Note Before Activating an Assembly

Remember the following points before activating an assembly:

- When creating assembly drawings, you must validate the components that have been imported from older versions of CADD5 or from other CAD systems through translators. Perform database checks on the components to ensure that the components are clean and that the drawing files are up-to-date for the following reasons:
  - A bad component can corrupt the database and can cause considerable loss of time while creating the drawings.
  - It is sometimes impossible to create HLR or section views using the bad components.
  - The drawing files for the models of the components that are not up to date can cause problems such as `segv` while creating detail drawings.
- Follow a corporate layer policy to ensure that only the geometry for the finished model is present on the specified layer for the complete model. Any extraneous data on this layer, or the other layers that are visible at the time of generating HLR images, can cause unpredictable results.

## Performing Database Checks

To perform database checks use the following steps:

1. Activate the component as a single part in the Explicit environment.
2. Use CHECK DBASE to file the component in the Explicit environment.

Release 13 Onward: You can use the `validate_db` utility for automatic database validation when activating or filing a part. To use this utility, set the following environment variables in the `.caddsrc` file to the path of the `validate_db` executable file:

- `DB_CHECKING`
- `DB_CHECKING_ACT`
- `DB_CHECKING_FILE`

Setting the `DB_CHECKING_ACT` environment variable in the `.caddsrc` file enables database validation when activating a part. Setting the `DB_CHECKING_FILE` environment variable enables database validation when filing a part.

If you have set the `DB_CHECKING` environment variable to the path of the `validate_db` executable, you can use the `ckCAD` utility for automatic database validation when activating or filing a part. If you have not set the `DB_CHECKING` environment variable, the `validate_db` utility is used for automatic database validation.

Revision 9 Onward: The HLR engine uses a matrix of the visible layers in all the selected views to identify the entities to be processed. Minimize the number of layers to be processed to reduce the HLR calculation time.

Revision 8: The entities that were processed by the HLR engine could be controlled by the `ECHO LAYER DRAW m, n-p` command. This command allowed only the entities on the visible layers of the drawing to be processed.

## Updating Drawings Using the `update_fig` Script

Use the `/cadds/script/update_fig` script to update all the figures in each CADDs part in the specified directory, to work with the new explicit Single Window Rendering (SWR) environment. Use the `-h` option for help on using the `update_fig` script.

The `update_fig` script automatically starts CADDs, activates each figure in the specified directory as an independent part, updates it for the new explicit SWR environment, and then files it. See, *Database Policies and Procedures* for details.

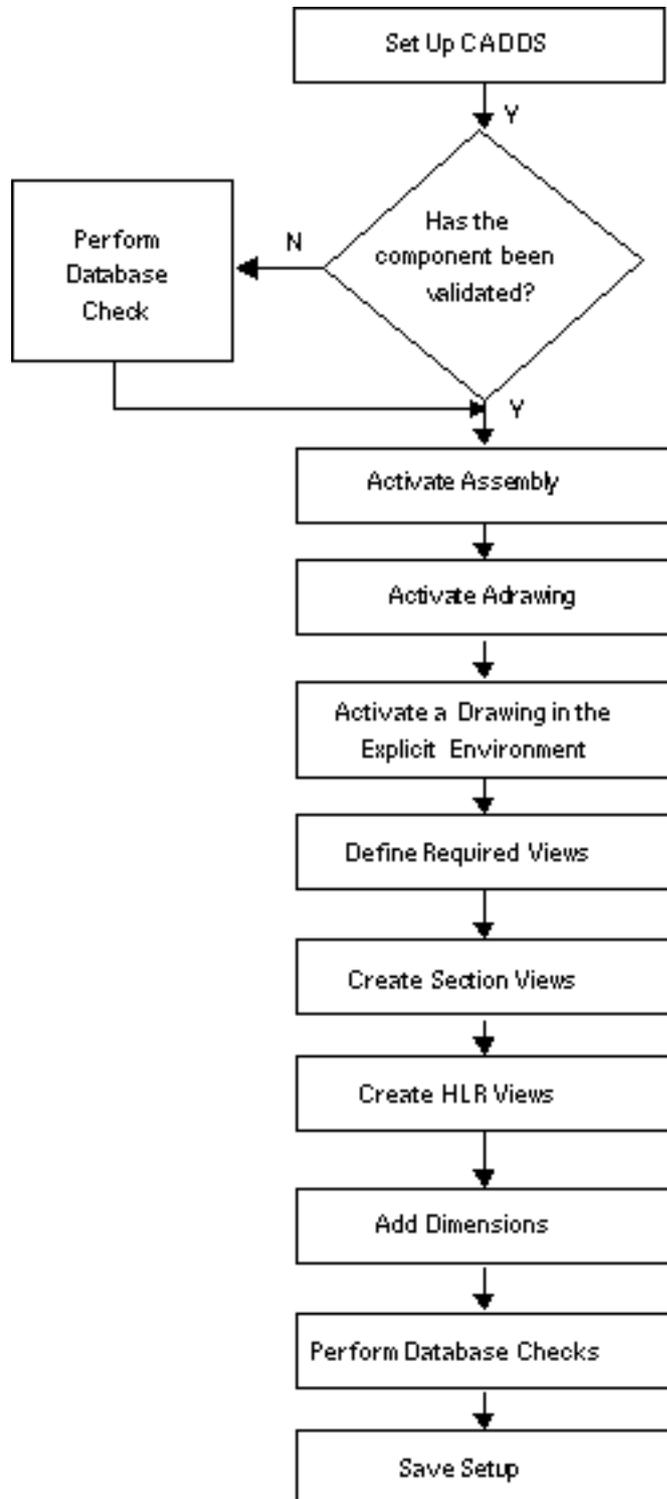
## Updating Drawings Using the `update_gr` Script

Use the `/cadds/scripts/update_gr` script to update the drawings of all the components of the assembly. For Help on using the `update_gr` script, use the `-h` option. The `update_gr` script automatically starts CADDs, activates each component in the assembly in the Part mode, and files it in the Explicit environment. This process synchronizes the drawings with the model of that component.

Please note: The `update_gr` script does not ensure that the components are cleaned up with respect to the database. As a precaution, perform database checks as mentioned earlier.

# Creating Assembly Detail Drawings

The following flowchart illustrates the best process of creating detailed drawings for assemblies. The details of each step are described in the relevant sections of this chapter.



## Setting Up CADDs

To obtain the best results during the detailing process for assemblies, CADDs requires the following setups:

- Setting up fonts
- Setting up parameter files
- Changing parameter files

For details see “Setting Up CADDs for Detailing” on page 2-3.

## Activating an Assembly

Activate an existing assembly only after ensuring that the assembly has been validated. For details on validating an assembly see “Validating an Assembly” on page 3-14. You can then activate the assembly in the Explicit environment by specifying the Adrawing name. To activate a new assembly, specify a new assembly name.

Please note: To validate an assembly, use CHECK DATABASE to perform database checks for the Adrawing.

After activating a model in CAMU, if you encounter the error,

```
ERROR 4 IN GENCMP CREATING DRAWING'S TVF'S
```

then perform the following steps.

1. Activate the model in Part mode.
2. Delete the existing MDRAW drawing.
3. File the part.
4. Reactivate the assembly. A new MDRAW drawing is created.

## Activating an Adrawing

Create all your assembly drawings in an Adrawing. An Adrawing is actually a CADDs part. The only difference between an Adrawing and a CADDs part is that when an Adrawing is filed in CAMU, the view states of the entire assembly tree are stored in the assembly database along with the Adrawing. Thus it is possible to store two different Adrawings for the same assembly with different view states of the assembly tree.

While activating an assembly, if you do not specify the name of the Adrawing, the Adrawing named `default` opens by default.

Please note: Do not use the default Adrawing for creating assembly drawings. Create a new Adrawing with a suitable naming convention. To distinguish parts from Adrawings, use the naming convention such that every Adrawing name starts with the prefix `ADRAW_<name>`.

## Working with Large Assemblies

When working with large assemblies, performance is an important consideration. After completing the layout of the drawing and creating the HLR and SECTION views in the drawing, you can open the Adrawing as a single part outside the assembly environment. You can then perform the dimensioning, add annotations, and modify the drawing, if required.

Please note: When you open an Adrawing as a single part, ensure that the NOTVF or UNLINKED options are not used when filing the part.

Revision 10 Specific: Dimensions that are associated between the local entity and the HLR entity may cause database corruption when UNHIDE OBJECT is applied to the view. Avoid creating such dimensions.

For Revision 9 and Release 11 Onward : Dimensions that are associated between the local entity and the HLR entity do not cause database corruption when you apply UNHIDE OBJECT to the view.

## Activating a Drawing in the Explicit Environment

Every time you activate a new Adrawing in the Explicit environment by default CADDs creates a drawing called DDRAW with the Adrawing. The DDRAW is a drawing within the CADDs part (Adrawing). Do not use the default DDRAW that is created with every Adrawing. Instead, create a new drawing with a suitable naming convention. To activate a new drawing, choose the ADRAWING option on the top bar of the Explicit assembly environment. Choose the DRAWING option from the resultant menu.

### Copying Similar Drawings

If two assembly drawings are similar, create two drawings within the same Adrawing. If there is little difference between the layout of the drawings, then you can create the second drawing by copying the first drawing using the COPY DRAWING command. This process reduces the time required to switch to a different Adrawing and to manipulate the view states of different components.

Please note: All HLR and section views are copied as normal views. There is no HLR or section information in the copied views.

Use COPY VIEW APPEARANCE to copy an HLR or section view across drawings within the same Adrawing.

Release 11 Onward: Copying the HLR or section views is possible in the Part mode as well as in the CAMU Assembly mode.

Release 10 Onward: Copying the HLR or section views is possible only in Part mode.

## Defining Required Views

An Adrawing provides a workspace on which you can view and manipulate viewed model instances. An Adrawing is both a CADD part and a drawing. The Adrawing contains views that allow you to see the model space with a particular orientation and scale that you define.

Adrawings allow you to define different views of the entire assembly, subassemblies, or specific components of the assembly. Each assembly can have many Adrawings, each with its own view state.

For details, see “Creating Drawing Layouts” on page 2-12.

## Creating Section Views for Assembly Drawings

The procedure to create section views in an assembly drawing is the same as that for creating sections in the part drawing. For details, see “Creating Section Views” on page 2-18. The following additional recommendations are specific to section views on an assembly drawing:

- Create a sectioner as a local entity of the Adrawing. Avoid deleting the sectioner. If you must delete the sectioner, first ensure that all the section views using the sectioner are cleaned up using the UNDEFINE SECTION command.
- Use DEFINE SECTION ALL to avoid the time-consuming task of manual selection of objects. DEFINE SECTION ALL selects all the entities visible in the view for sectioning. Therefore, before you issue DEFINE SECTION ALL, ensure that all the required objects are viewed and are visible in the view.
- Avoid using the WIN option that is available from the top bar of the CADD desktop for selecting all the objects visible in the view. Instead, use DEFINE SECTION ALL or use DEFINE SECTION with the VWIN option that is available from the top bar of the CADD desktop to select the sectionees.
- In the Assembly environment, objects that are being sectioned are not blanked automatically as in the case of HLR. If you want to see only the section results in the section view, use the following command to define the section:

```
DEFINE SECTION [modifiers] SECTNAME <section-name> LAYER  
<layer-num>: .
```

This command creates a separate view with only the specified layer number echoed in the section view. Thus, only the sectioning results are visible in the section view.

Revision 8 Specific: Use BLANK COMPONENT to blank the unwanted entities in the section view so that DEFINE SECTION LAYER produces the required results.

- When the section views are created in an Adrawing, the size of the database for the Adrawing increases. Because the components are viewed in an Assembly environment, CADD uses a different mechanism for storing section-related information in the database for CAMU. For creating a section of viewed components, CADD creates surrogate objects in the database of the Adrawing.

Release 13 Specific: If the sectioner is deleted, you can use REGENERATE SECTION to select a new sectioner. Select only the local entity as a sectioner. The section view is regenerated with the new sectioner.

Please note: Release 13 onward, you can also use a plane as a sectioner.

# Creating HLR Views on Assembly Drawings

The procedure to create HLR views in the assembly drawing is the same as that for the part drawing. For details, see “Creating HLR Views” on page 2-32.

## Working with HLR Views

Additional recommendations specific to working with HLR views on an assembly drawing are as follows:

- Use **HIDE OBJECT ALL** to avoid the time-consuming task of manual selection of objects. The **ALL** modifier automatically selects all the entities visible in the HLR view. Therefore, before you issue the **HIDE OBJECT** command, ensure that all the required objects are visible in the view.
- Avoid using **HIDE OBJECT WIN** for selecting all the objects that are visible in the view. Instead, use **HIDE OBJECT ALL** or **HIDE OBJECT VWIN** to select the objects on which HLR will be performed.
- Avoid using the **MOVE COMPONENT** and **ROTATE COMPONENT** commands while creating assembly drawings. When you perform assembly operations such as moving and rotating components, the surrogates move along with the components and HLR views in the drawing are disturbed.
- When you use **HIDE OBJECT ALL**, blanked entities get selected and you may not get the desired results. Before performing HLR, move the unwanted geometry to a layer that is not visible in the current view instead of blanking the entities.

Please note: When you perform an HLR operation in an Adrawing, the surrogate objects and surrogate curves are created in the database of the Adrawing. Thus, when you create HLR views, the database size of the Adrawing increases.

## Creating Exploded HLR Views

To create the exploded HLR views use the following procedure:

1. Change the assembly from the standard to the exploded view.
2. Use **HIDE OBJECT NONASSOC** to create an exploded HLR view.
3. After the HLR image is created, change the assembly back to the standard view.
4. The standard HLR views can be created by using **HIDE OBJECT**, which creates an associative HLR image.

## Validating an Assembly

Use CHECK DATABASE to perform database checks for the Adrawing after you have created the HLR and section views. For details see “Checking the Validity of Components” on page 2-8.

# Adding Dimensions, Centerlines, or Annotations to Assembly Drawings

You can directly add associative dimensions, centerlines, or annotations to viewed components from Revision 8 onward. When HLR is applied to a viewed component, the dimensions associated with the viewed component are transferred to the HLR entities and vice versa. Thus, in the CAMU environment, you can add dimensions or centerlines to the viewed components directly.

Please note: You can add only explicit dimensions or centerlines to the viewed components.

When working with large assemblies, add dimensions or centerlines outside the assembly environment by activating the Adrawing as a part as explained in the section “Working with Large Assemblies” on page 3-9. The dimensions that are associated to HLR views outside the assembly environment are still associative when the Adrawing is activated again in the assembly environment.

## Saving the Setup and Filing the Adrawing

The procedure for saving the setup and filing the Adrawing are the same as that for parts.

For details, see section “Filing the Part for Saving the Drawing” on page 2-48. After you have finished creating the drawing, you must save the Adrawing. It is not necessary to save the entire assembly at this time.

## Updating the Assembly Detail Drawing

This section discusses how to update an Adrawing from CADDs Revision 8 and Revision 9. From Revision 10 onward, you are automatically informed whenever a drawing is due for update.

Please note: If the section or HLR views have components that are viewed off or are closed, you are not given any indication that the views require updating. You must first view on these components before attempting to update them. Otherwise, you can lose relevant HLR information.

### Updating the Drawing for Minor Changes in the Assembly

The process to update an assembly that has undergone minor changes follows. The details regarding these steps are described in the subsequent sections.

1. Ensure that the assembly is clean with respect to the database.

Please note: Use `BLANK COMPONENT` or `UNBLANK COMPONENT` instead of `VIEW COMPONENT ON` or `VIEW COMPONENT OFF`. The `BLANK COMPONENT` and `UNBLANK COMPONENT` commands enable you to keep track of the visible components more easily.

2. Use `VIEW COMPONENT ON` to view all the components that have been used by HLR and section views.
3. Update all the section views.
4. Update all the HLR views.
5. Perform database checks and file the Adrawing.
6. Define additional HLR and section views, if required.
7. Add dimensions to sections again, if required.
8. Define any additional dimensions or centerlines, if required.

Please note: To add dimensions to sections or to add any dimensions outside the CAMU environment, open the Adrawing as a single part.

## Cleaning the Database

Ensure that the assembly is in a clean state before updating the assembly drawing. You must ensure that components that have undergone changes are not corrupt before you update any drawing. Perform database checks as discussed in the section “Performing Database Checks” on page 3-3. The **REGENERATE SECTION** or the **UPDATE HLRIMAGE** commands do not produce the required results if a component is corrupt. A corrupt component can cause a `segv` error or can corrupt the Adrawing.

## Viewing the Components

You must use **VIEW COMPONENT ON** for all the components that have undergone HLR or sectioning. Using **VIEW COMPONENT ON** for the entire tree ensures that all the components present in the drawing are visible. This time-consuming task is worth the effort for very large trees.

- If the components used for HLR are not viewed on, **UPDATE HLRIMAGE** does not update the curves of the unviewed objects. Only the viewed objects pass through the HLR engine.
- If some of the sectionees are viewed off while you apply the **REGENERATE SECTION** command, **CADDS** may delete the section view and leave garbage entities in the database resulting in a `segv` error.

Please note: Components that are blanked by using **BLANK COMPONENT** are picked up by the **REDEFINE SECTION** or the **UPDATE HLRIMAGE** commands. This action does not cause any problems.

## Updating Section Views

Guidelines for updating section views follow:

Revision 9 Onward : Use **REGENERATE SECTION ALL** to update all the section views in a drawing simultaneously. If the sectioner or the definition view has been deleted, **REGENERATE SECTION** may fail and not clean the section view. Use **UNDEFINE SECTION** to clean the view. Then redefine the section view using **DEFINE SECTION**.

Revision 8 Specific : First clean up all the cutaway sections on the drawing using the **UNDEFINE SECTION** command. Then use **REGENERATE SECTION ALL** to update the rest of the section views on the drawing.

## Updating HLR Views

Use `UPDATE HLRIMAGE` to update all the HLR views. If some components have been removed or the model has been changed, the HLR curves corresponding to the component are cleaned up when you use `UPDATE HLRIMAGE`.

## Updating the Drawing for Major Changes in the Assembly

If the assembly has undergone major changes, undefine all the sections and unhide all the HLR views. You must then redefine the HLR and section views.



# Enhancements in Drawing Generation

---

This appendix lists the enhancements in the drafting and drawing generation functionality for specific revisions from Revision 8.

- Revision-Specific Enhancements

# Revision-Specific Enhancements

The following table lists the enhancements for various CADDs revisions, from Revision 8. The asterisk (\*) indicates the release for which the enhancement was introduced. These enhancements are available for all future releases.

**Table A-1 Revision-Specific Enhancements**

Functionality	Feature	R8	R9	R10	R11	R12	R13	R14	R15
Context Sensitive Dimensioning	Introduced this functionality	*							
	Hot keys		*				*		
	Insertion of linear dimension		*						
	Insertion of radial dimension		*						
	Insertion of diameter dimension		*						
	Insertion of angular dimension		*						
	Dynamic dragging of text		*						
	Alignment of two dimensions		*						
	Modification of the dimension properties		*						
	Separate tolerance ON/OFF toggle for primary and dual dimensions				*				
	Addition of an option to add a bracket in primary or secondary dimension text individually				*				
	Option to select a decimal separator for secondary dimension text.								*
	Introduction of primary and secondary dimensions to track the units of tolerance automatically				*				
	Dynamic modification of extension line gap				*				
	Introduction of snapping of the model and draw grid in the CSD environment				*				
	Enabling of <code>getdata</code> filters such as <code>intof</code> and <code>tangent</code> in CSD				*				
	Introduction of editing of text on FCS: Shows the actual symbol instead of special characters				*				
	Addition of the faceted engine to HLR and the new menu user interface for HLR				*				
	Primary and secondary tolerance track each other					*			
	The MODIFY FCS command in CSD brings up ANSI menu when in ISO mode					*			

**Table A-1 Revision-Specific Enhancements**

Functionality	Feature	R8	R9	R10	R11	R12	R13	R14	R15
	Options for aligning and dragging multiple dimensions					*			
	Regeneration of intersection dimensions					*			
	Ability to attach URLs or References to Explicit Text and Labels, and launching the URLs through a browser						*		
	Dynamic dragging of Cnodes, Tnodes, and Nfigs using CSD in Draw mode						*		
	Creation of dimensions between intersections of general curves						*		
	Editing the parameters of annotations within CSD						*		
	Insertion and editing of annotations in CSD of detailing entities such as balloon, finish symbol, datum symbol, datum target, and weldmark to support associativity						*		
	Dynamic dragging of annotations in CSD						*		
	Provision to insert associative multiple arrowhead label							*	
	Dragging of segments of a single or multiple associative or non-associative arrowhead labels							*	
	Dragging of vertices of a single or multiple associative or non-associative arrowhead labels							*	
	Dragging of arrowhead or the entire label along the entity with which it is associated.							*	
Vertex Associative Dimensioning	Preservation of dimension associativity on topological changes to solids			*					
Dimension Update Management	Automatic update indications for drawings in the single part			*					
	Automatic update indications persistent across part activations					*			
	Automatic update indications for assemblies (CAMU)					*			
Associate Dimensions to the Sections and HLR									
Copy View	Allow copy of the HLR or section view in Single Part mode		*						

**Table A-1 Revision-Specific Enhancements**

Functionality	Feature	R8	R9	R10	R11	R12	R13	R14	R15
	Introduction of Copy View Appearance to copy HLR and section views in the parts and CAMU environment			*					
HLR and Sectioning	Preservation of fonts by HLR commands		*						
	HLR support of intersection curves		*						
	Performing HLR on only those entities present within or crossing the selected view boundaries, that is, the CLIP option							*	
	Fixed the database corruption on regeneration of cutaway section views		*						
	Introduction of new modifiers to UPDATE HLRIMAGE				*				
	Introduction of the MERGE modifier to give local control on the HLR image during update				*				
	Enabling or disabling the tangent edges without running the engine				*				
	Enabling or disabling the Topological edges without running the engine.				*				
	Controlling the quality and performance using the slide bar				*				
	Displaying the hidden lines as blank or dashes without running the engine				*				
	Selecting the engine at the time of update				*				
	Provision for detailed status of an HLR view				*				
	Define planes as section tools						*		
	Ability to redefine sections if sectioner is deleted						*		
	Clip In and Clip Cross options							*	
	Mark Include and Mark Exclude options							*	
	Dependency of HLR on the standard or exploded mode of the assembly							*	
Associative Detailing Entities	Implementation of detailing entities such as balloon, finish symbol, datum symbol, datum target, and weldmark to support associativity						*		

**Table A-1 Revision-Specific Enhancements**

Functionality	Feature	R8	R9	R10	R11	R12	R13	R14	R15
	Provision to add up to nine text labels in a weldmark symbol							*	
Automatic DB Validation	Automatic database validation using the <code>validate_db</code> utility when activating and filing a part						*		



---

# Index

---

## A

- Activating
  - Part 2-10
- Activating an Adrawing 3-8
- Activating an Adrawing in the Explicit Environment 3-10
- Activating an Assembly 3-8
- Adding Dimensions and Centerlines to Assembly Drawings 3-15
- Aligning
  - Two Views 2-13
- Annotation
  - Title Blocks 2-5
- Assembly
  - Activating 3-8
  - Activating a Drawing in Explicit 3-10
  - Activating an Adrawing 3-8
  - Adding Dimensions and Centerlines 3-15
  - Cleaning the Database Before Update 3-18
  - Copying Similar Drawings 3-10
  - Creating Detail Drawings Flowchart 3-5
  - Creating Exploded HLR Views 3-13
  - Creating HLR Views on Assembly Drawings 3-13
  - Creating Section Views for Assembly Drawings 3-12
  - Defining Required Views 3-11
  - Drawings
    - Setting Up CADDs 3-7
  - Drawings Update Using the update\_gr Script 3-4
  - Filing the Adrawing 3-16
  - Performing Database Checks 3-3
  - Saving the Setup 3-16
  - Updating Drawing for Major Changes 3-19

- Updating Drawing for Minor Changes 3-17
- Updating HLR Views 3-19
- Updating Section Views 3-18
- Updating the Detail Drawing 3-17
- Validating 3-14
- Viewing Components 3-18
- Working with HLR Views 3-13
- Working with Large Assemblies 3-9

## C

- Change
  - Settings in the parameter file 2-4
- Changing the Layout
  - Moving Views 2-16
- Checking
  - Validity of Components 2-8
- Configuring
  - Memory Requirement for HLR 2-7
    - Errors for Insufficient Space 2-7
    - Temporary File Space 2-7
- Creating
  - Centerlines 2-28
    - Conventional Method 2-28
    - Defining Features 2-29
    - Parametric Method 2-28
  - Design Dimensions 2-42
    - Advantages 2-41
    - Disadvantages 2-41
    - Parametric Approach 2-41
  - Detail or Scrap Views 2-17
  - Dimensions 2-39
    - Associative 2-39

- Tips 2-40
- Drawing Layouts 2-12
- Explicit Dimensions 2-39
- HLR
  - Analytical Engine 2-35
  - COPY VIEW APPEARANCE 2-32
  - Ensuring Quality 2-33
  - Faceted Engine 2-34
  - HIDE OBJECT REFINE 2-35
  - HIDE OBJECT TAN 2-32
  - Unhiding 2-36
  - Updating 2-35
- HLR Views 2-32
- Section in a separate view 2-21
- Section in an existing view 2-23
- Section Views 2-18
- Sections
  - Points to Note 2-25
- Simple Section View 2-22
- Unfolded Section 2-22
- Creating Assembly Detail Drawings 3-5
- Creating HLR
  - HIDE OBJECT FACETED 2-34
- Creating HLR Views on Assembly Drawings 3-13
- Creating Section Views for Assembly Drawings 3-12
- Creation
  - Detailed Drawing for Parts 2-2
  - Explicit New Drawing Forms 2-4
- Customizing
  - Menus 2-6

## D

- Design Dimensioning
  - Solutions to issues
    - Corrupt Arrowheads 2-47
    - Cplane Orientation 2-47
    - Sketcher parameters 2-47
- Design Dimensions
  - Creating 2-42
  - Managing 2-44
  - Solutions to issues 2-47
  - Using 2-41
- Dimensions
  - Creating 2-39
    - Tips 2-40

- Documentation, printing from Portable Document Format (PDF) file 1-x
- Drawing
  - Creation 1-3
- Drawing Generation
  - Overview 1-2
  - Part
    - Activating 2-10
    - Activating a New Drawing 2-11
    - Adding Dimensions 2-39
    - Aligning Two Views 2-13
    - Annotating the Title Blocks 2-5
    - Checking Validity of Components 2-8
    - Configuring memory requirement 2-7
    - Creating a Simple Section View 2-22
    - Creating an Unfolded Section 2-22
    - Creating Centerlines 2-28
    - Creating Design Dimensions 2-42
    - Creating Detail Views 2-17
    - Creating Detailed Drawings 2-2
    - Creating Drawing Layouts 2-12
    - Creating Explicit Dimensions 2-39
    - Creating HLR image
      - Analytical Engine 2-35
      - Faceted Engine 2-34
    - Creating HLR Views 2-32
    - Creating New Explicit Drawing Forms 2-4
    - Creating Scrap Views 2-17
    - Creating Section in Existing View 2-23
    - Creating Section in Separate View 2-21
    - Creating Section Views 2-18
    - Creating sections
      - points to note 2-25
    - DEFINE SECTION 2-18
    - Defining Features
      - Creating Centerlines 2-29
    - Design Dimensioning
      - Solving problems 2-47
    - Design Dimensions 2-41
    - Ensuring Quality of the HLR Image 2-33
    - Filing the Part 2-48
      - SAVE SETUP 2-48
    - Managing Design Dimensions 2-44
    - Moving All Views 2-17
    - Moving Views 2-16
    - Saving the Drawing 2-48
    - Setting up a General Drawing Layout 2-12

- Setting up CADDs 2-3
- Setting Up Fonts 2-3
- Setting Up the Parameter File 2-3
- Unhiding the HLR View 2-36
- Updating a Drawing for Major Modeling Changes 2-51
- Updating a Drawing for Minor Modeling Changes 2-50
- Updating a Section View 2-26
- Updating an HLR View 2-35
- Updating Detail Drawings 2-50

- Process 1-3
  - Creating Drawings 1-3
  - Updating Drawings 1-4

## E

- Errors
  - Insufficient Space 2-7
- Explicit New Drawing Forms
  - Creation 2-4

## F

- Filing
  - Part 2-48
- Filing the Adrawing 3-16

## H

- HLR Image
  - Quality 2-33

## M

- Moving
  - All Views 2-17
  - Views 2-16

## O

- Overview
  - Drawing Generation 1-2

## P

- Parameter File Settings
  - Change 2-4
- Performing Database Checks 3-3
- Printing documentation from Portable Document Format (PDF) file 1-x
- Process
  - Drawing Generation 1-3

## S

- SAVE SETUP
  - Saving the drawing 2-48
- Saving
  - Drawing 2-48
- Section Views
  - Creating 2-18
- Setting
  - General Drawing Layout 2-12

## T

- Title Blocks Annotation
  - Using Tnodes 2-5

## U

- Unhiding
  - HLR 2-36
- update\_gr Script 3-4
- Updating
  - Assembly Drawings Using update\_gr Script 3-4
  - Drawing for major modeling changes 2-51
  - Drawing for minor modeling changes 2-50
  - HLR View 2-35
  - Section View 2-26
    - Revision 8 2-27

Updating the Assembly Detail Drawing 3-17

Updating the Drawing for Major Assembly  
Changes 3-19

Updating the Drawing for Minor Assembly  
Changes 3-17

Using

    DEFINE SECTION 2-18

        Advantages 2-18

        Capabilities 2-18

    Design Dimensions 2-41

## V

Validating an Assembly 3-14

Viewing the Components 3-18

## W

Working with Large Assemblies 3-9