Appearance and Hidden Line Removal Reference

CADDS® 5 15.0

DOC36800-012

Parametric Technology Corporation

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

Commands

User Aids

System Messages

Basic Procedure File Shapes

Preface

Appearance and Hidden Line Removal Reference describes commands that compile procedure files, create three-dimensional architectural renderings, and produce detailed drawings. This book is intended for Architectural and civil engineers, architects, interior designers and architectural drafters who are familiar with computer-aided-design (CAD).

Related Documents

The following documents may be helpful as you use *Appearance and Hidden Line Removal Reference*:

- H*eating,Ventilating and Air Conditioning Reference*

  This book discusses the HVAC commands used for modeling of duct runs, determining heating and cooling loads, and creating input data files for heating and cooling load calculations.

- *Visualization/Preparation Reference*

  This book describes Visualization commands, and commands used to automatically create procedure files (CONSTRUCT PROCEDURE, CONSTRUCT EQUIPMENT, and PREPARE EQUIPMENT).

- *Piping Reference*

  This book provides an overview of the modeling and drawing figures used for piping design, as well as information on supporting files.

- *Explicit Modeling User Guide and Menu Reference*

  This book provides a complete overview of the CADDS graphic system, including general features and uses. It also provides all the information you need for CADDS concepts, vocabulary, punctuation syntax and error correction procedures.
- *Hidden Line Removal (and AEC HLR) User Guide and Menu Reference*
- This book provides information on how to blank the features on a solid model that would be invisible or hidden from the viewer if the object were solid

Book Conventions

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

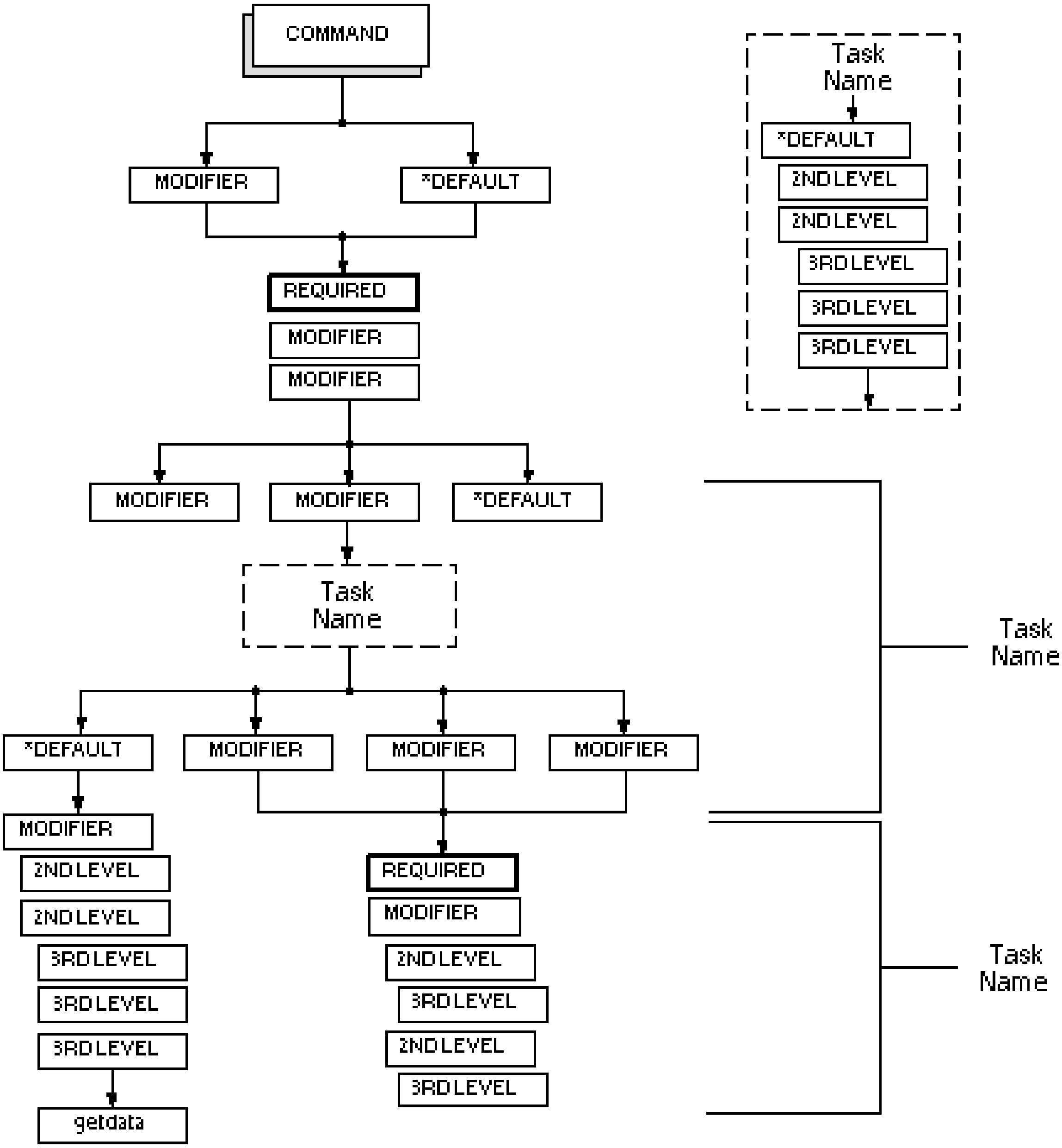
Convention	**Example**	**Explanation**
Menu selections and options	List Section option, Specify Layer field	Indicates a selection you must make from a menu or property sheet or a text field that you must fill in.
User-selected graphic location	X, d_1 or P1	Marks a location or entity selection in graphic examples.

User input in CADDS text fields and on any command line	`cvaec.hd.data.param`
	`tar -xvf /dev/rst0`
		Enter the text in a CADDS text field or on any command line.

System output	`Binary transfer complete.`	Indicates system responses in the CADDS text window or on any command line.
Variable in user input	`tar -cvf /dev/rst0` filename	Replace the variable with an appropriate substitute; for example, replace filename with an actual file name.
Variable in text	tagname	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	INSERT LINE TANTO	Shows CADDS commands and modifiers as they appear in the command line interface.
Text string	`"SRFGROUPA"` or `'SRFGROUPA'`	Shows text strings. You must enclose text string with single or double quotation marks.
Integer	n	Supply an integer for the *n.*
Real number	x	Supply a real number for the *x*.
#	`# mkdir /cdrom`	Indicates the root (superuser) prompt on command lines.
%	`% rlogin` remote_system_name `-l root`	Indicates the C shell prompt on command lines.
$	`$ rlogin` remote_system_name `-l root`	Indicates the Bourne shell prompt on command lines.

Modifier Diagrams

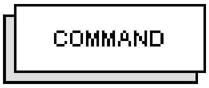
Modifier diagrams are used in this book to show the flow of a command and its modifiers. Diagram conventions are explained in the next section.



Conventions

Modifier diagram conventions are shown below. Modifier diagrams do not show where you must press the RETURN key or enter punctuation, such as a semicolon.

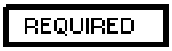
This information is provided in the syntax section for each command. You need not enter all modifiers in a path to get to the end of that path.



Commands are listed in the shadowed box.

COMMAND
MODIFIER

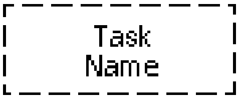
Commands and modifiers are capitalized.

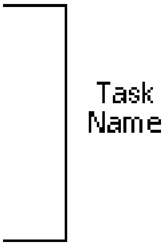


Required modifiers are in bold boxes.

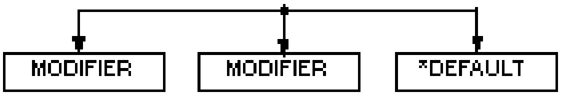
An asterisk (*) denotes a default modifier.

Use of getdata is shown in a box.

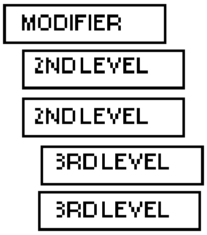




Dashed boxes and brackets group modifiers related to a task. The name of the task appears in the box or beside the bracket. If you do not wish to perform that task, you will not need to use any of the modifiers in that group.



Mutually exclusive modifiers are shown horizontally.



Second and third level modifiers are indented under the corresponding first level modifier.

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 CADDS desktop or the Local Data Manager (LDM)

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

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/cadds/data/html/htmldoc/` (UNIX)

   Drive:`\usr\apl\cadds\data\html\htmldoc\` (Windows)

2. Click `mainmenu.html`. A list of available CADDS documentation appears.
3. Click the book title you want to view.

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 DOCUMENTATION. A list of available CADDS 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 CADDS desktop in the following ways:

- From the Information Access button on the CADDS 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`.

Chapter 1

Commands

This chapter describes, in detail, different commands that allow you to compile procedure files, create three-dimensional drawings, and produce detailed drawings.

- Overview
- Command Format

Overview

The following commands allow you to compile procedure files, create three-dimensional drawings, and produce detailed drawings:

- COMPILE PROCEDURE
- CREATE RENDERING
- EDIT APPEARANCE
- GENERATE DETAIL
- GENERATE DLINE
- GENERATE SURFACE
- HIDE PIPE
- MERGE RENDERING
- UNHIDE PIPE
- HIDE OBJECT
- UPDATE HLRIMAGE

Compiling Procedure Files

The COMPILE PROCEDURE command compiles procedure files you have created manually with a text editor or automatically with the CONSTRUCT PROCEDURE (see *Visualization/Preparation Reference*) command. Procedure files are text files that use basic shapes (such as boxes and cones) to define three-dimensional detail, clearance, and surface representations of components. These basic shapes are illustrated and described in Appendix C, “Basic Procedure File Shapes”.

Compiled procedure files are used by the GENERATE DETAIL, GENERATE DLINE, GENERATE SURFACE, and CHECK CLEARANCE (see the *Reports and Analyses Reference*) commands.

You can also use the CONSTRUCT EQUIPMENT and PREPARE EQUIPMENT commands to create procedure files. CONSTRUCT EQUIPMENT models a piece of equipment using the basic shapes. PREPARE EQUIPMENT prepares the model nodal figure as a figure representation, and creates and compiles procedure files that define the detail and clearance representations.

Please note: It is not necessary to use the COMPILE PROCEDURE command when you create procedure files with CONSTRUCT EQUIPMENT and PREPARE EQUIPMENT; it is done automatically. For more information on these commands, see the *Visualization/Preparation Reference*.

Creating Three-Dimensional Drawings

You can create realistic three-dimensional drawings with hidden line removal from models created with nodal figures (Nfigures), nodal lines (Nlines), walls, and projected outlines.

You create Nfigures for design components such as furniture, equipment, and structures by building them with basic shapes, creating a procedure file, and preparing them as Nfigures. For example, a figure that is to become a three-dimensional rectangular table with a square base and a round pedestal consists of a rectangular box, a square box, and a cylinder. The way the shapes are put together is stored as a procedure file and the part is stored as an Nfigure. You can then use the Nfigure to build other models that can be rendered.

Use the following commands to create three-dimensional drawings from Nfigure models:

1. CONSTRUCT PROCEDURE

Create a procedure file (text file) for the defined shapes. (For more information, see the *Visualization/Preparation Reference*.)

2. EXIT PART FILE NFIGURE

File the part, preparing the corresponding nodal figure files.

3. ACTIVATE PART partname

Activate the part for the final design.

4. INSERT NFIGURE

Insert the Nfigures into the data base.

5. SELECT RELATIONSHIP ON

Relate original Nfigures to their rendered shapes (if a second generation rendering).

6. CREATE RENDERING

Erase hidden lines.

7. EDIT APPEARANCE

Touch up the completed drawings.

You can use CREATE RENDERING on nodal lines (Nlines) with the DUCTSIZE or OUTERDIAM properties; these properties are added by the ROUTE DUCT (see the *Heating, Ventilating, and Air Conditioning Reference*) or ROUTE PIPE (see the *Piping Reference*) commands, respectively. You do not need to prepare an Nfigure or a procedure file. The rectangular box shape (for ducts) or the cylinder shape (for pipes) is used for hidden line removal.

Renderings can be modified, such as adding or deleting furniture to an interior design. You can erase hidden lines interactively for Nfigures rendered in separate executions of CREATE RENDERING. To erase hidden lines interactively, use the MERGE RENDERING command. To hide lines in selected regions of the drawing, use the EDIT APPEARANCE command.

Creating Renderings with Perspective

Designs can also be rendered in perspective. First create and prepare the Nfigures, insert them, execute DEFINE PERSPECTIVE to generate a wire frame perspective, and finally, use the CREATE RENDERING command to remove the hidden lines. For more information on DEFINE PERSPECTIVE, see the *Core CADDS Reference.*

Producing Detailed Drawings

Appearance and hidden line removal commands change the visual appearance of pipes, ducts, in-stream components, and major equipment or structural items to produce single- or double-line detailed drawings. Before using GENERATE DETAIL and GENERATE SURFACE, you must

1. Create procedure files that define the solid, three-dimensional representations of components in terms of basic shapes such as cones, cylinders, and boxes. You must prepare these procedure files for all of your equipment and any fittings that you add to the standard CVPD library.
2. Use the COMPILE PROCEDURE command to put procedure files in a form that the GENERATE DETAIL and GENERATE SURFACE commands can use.

GENERATE DETAIL automatically generates single- or double-line detail graphics (wire frame graphics with hidden lines removed), and GENERATE SURFACE creates surface entities.

INSERT PBREAK (see the *Piping Reference*) creates pipe break details for single- and double-line pipes.

With the HIDE PIPE command, you can do some final touching up where detailed components hide other components that were detailed in separate executions of the command. To create a three-dimensional appearance, the detail graphics that would be hidden are blanked. You can also use the command to produce a cutaway view. The UNHIDE PIPE command re-displays hidden lines, returning a component to its original appearance. This command is particularly useful in making cutaway views or changing hidden lines affected by changes in the modeling layout. The HIDE INTERSECTION command is useful for touching up intersections of piping and instrumentation lines in P&IDs. For more information on HIDE INTERSECTION, see the *Piping Reference*.

After a duct schematic is completed, you can use the GENERATE DLINE command to display a double-line or triple-line representation of the duct run and fittings. For the orthographic view, a double-line representation is displayed. For isometric or perspective views, a triple-line three-dimensional representation is displayed. For information on other commands used to model duct runs, see the *Heating, Ventilating, and Air Conditioning Reference*.

Command Format

This section uses the following format conventions to describe each command:

Description	**Explains the general purpose and use of the command.**
Valid Modes	Specifies the mode(s), Model and/or Draw, in which you can use the command.
Syntax	Consists of two parts: (2) the verb/noun combinations and modifiers and, in most cases, (2) getdata information, which is the location information and/or entity identification.
Modifiers	Lists and describes the valid modifiers for the command and specifies any default modifier values. Included in this section are modifier diagrams. Using modifier diagrams is explained in Notes to Readers. The MODFILE modifier is not listed with each command in this section, although it is valid with almost all commands.MODFILE filename incorporates the components of the modifier file filename into the command. These components are executed in sequence after the modifier that precedes MODFILE, if any. For more information, see the *Explicit Modeling User Guide and Menu Reference.*

COMPILE PROCEDURE

Description

This command compiles procedure files that you create. Procedure files are text files that use basic shapes to define three-dimensional detail, clearance, and surface representations of components. These representations are used by the GENERATE DETAIL, CHECK CLEARANCE (see the *Reports and Analyses Reference*), and GENERATE SURFACE commands.

GENERATE DETAIL, CHECK CLEARANCE, and GENERATE SURFACE determine which procedure file to use from the procedure name (PNAME) property inserted on the component. If none is present, the Nfigure name is used. In either case, this name is the same as that of the compiled procedure file minus the _PROC. The commands get the parameters as follows:

1. If the component has a PLIST property, the values in this list are used.
2. If the component has a STOCKNO property, the stock number is used to look up the values in the active parameter file.

If both these properties exist, PLIST overrides STOCKNO. If neither property exists, all the necessary parameters must be defined directly, with constants and/or component scales in the procedure.

Appendix C lists and illustrates all the basic shapes supported by COMPILE PROCEDURE, GENERATE DETAIL, CHECK CLEARANCE, and GENERATE SURFACE. Appendix C also provides some information on defining shapes in a procedure file.

Valid Modes

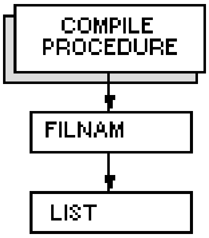
All modes

Syntax

```
#n# COMPILE PROCEDURE filename [modifiers]
```

Modifier

The following diagram shows the flow of COMPILE PROCEDURE modifiers. For a full explanation of modifier diagrams, see Notes to Readers.



Modifiers are listed alphabetically by level. When there are multiple levels of modifiers, level is indicated by the left number. A description of each modifier follows.

1. FILNAM filename

Specifies the name of the procedure file to compile. (The &BCD is omitted.) This modifier is optional, if the filename is entered immediately after COMPILE PROCEDURE.

2. LIST

Lists the entire procedure file and any diagnostic checking during compilation. Without this modifier only faulty lines and system messages are displayed.

Procedure

```
#n# COMPILE PROCEDURE filename [modifier]
```

1. Enter the COMPILE PROCEDURE command.
2. Then enter the name of the text file to compile.
3. Press RETURN.

Examples

Three examples of the COMPILE PROCEDURE command are shown. The first example includes errors to illustrate system messages. The second example has no errors and shows only a listing without messages. The third example shows the same procedure compiled without the LIST modifier.

If the input has errors, compilation fails and no secondary (compiled) file is produced. However, the detail generation commands will run if a previous version of the file was successfully compiled.

Example 1

```
#n# COMPILE PROCEDURE CVPD.F.EX1 LIST
SHAPE=CONX,L=T3yT2,D1=T1,D2=.5*(T4+T1)
      o
****     UNRECOGNIZABLE SHAPE     ****
SHAPE=CYL,,L=T2,D1=T4
      o
****     ----- SYNTAX
CLEARANCE
SHAPE=CYL,DD=T4,L=T3
      o
****     UNRECOGNIZABLE KEYWORD     *****
```

Example 2

```
#n# COMPILE PROCEDURE CVPD.F.EX2 LIST

SHAPE=CONE,L=T3-T2,D1=T1,D2=.5*(T4+T1)
SHAPE=CYL,L=T2,D1=T4
*
CLEARANCE
SHAPE=CYL,D1=T4,L=T3
*
```

Example 3

This example shows a procedure for a tube fitting with millimeter units.

The procedure uses millimeter units, but it recognizes that the fitting part must be modeled in inch units for compatibility with the insertion software. It does not matter what the data base units of the model are into which the fitting is inserted. SX and SY are in inches, so they are multiplied by the constant 25.4 and converted to millimeters. In the line defining the cone, an offset of X = T4yT3 is defined. If the value of 2 was not multiplied by T8, it would be interpreted in data base units of meters. This offset is needed to ensure the proper hidden line removal between the cone and diffusers.

Note that the pyramidal concentric diffusers (PCDF) are oriented in opposite directions along the y-axis by using DY = +1 versus DY = y1. When the fitting is inserted into the model with its y-axis up, the pyramids are oriented up/down, as shown below.

Also notice the use of the trigonometric functions. This is shown as an example. Since these are functions of known angles, it would be better to enter the actual numbers. Then every time the procedure is used, the system need not calculate the value of the trigonometric functions.

The use of SX and SY in the procedure is also unnecessary. The need to convert these values from inches to millimeters could have been avoided by using additional parameters. The fitting library uses this approach by having T1=SX and T2=SY. This example was prepared so that the substitution of T1 and T2 for 25.4*SZ and 25.4*SY in the procedure file and the substitution of 10 and 50 for the zeroes in the PLIST would make it consistent with the library approach.

The following command compiles the procedure without listing it:

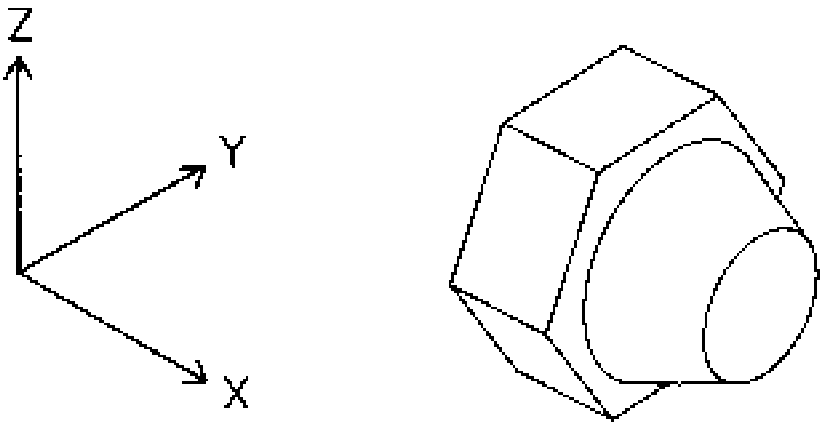
```
#n# COMPILE PROCEDURE EX3.TUBEFIT NOLIST

UNITS=MM
* SX  =     DISTANCE FROM FACE OF TUBING TO FACE OF THREADED PIPE
* SY  =     NUT SIZE — FLAT TO FLAT
* T3  =     THREAD ENGAGEMENT
* T4  =     NUT THICKNESS
* T5  =     CONE LENGTH
* T6  =     CONE DIAMETER AT NUT END (LARGER DIAM)
* T7  =     CONE DIAMETER AT TUBE INLET END
*
SHAPE = PCDF,L=0.5*25.4*SY, H=T4, HE=T4, DI=25.4*SY/COS(30),
*           D2=TAN (30)*25.4*SY, X=0.5*T4yT3, DY=+1, EDZ=1
SHAPE = PCDF,L=0.5*25.4*SY, H=T4, HE=4, D1=25.4*SY/COS (30),
*           D2=TAN (30)*25.4*SY, X=0.5*T4yT3, DY=1, EDZ=1
```

```
SHAPE = CONE,L=T5, D1=T6, D2=T7, X=T4yT3, DX=1
CLEARANCE
SHAPE = CYL, L=SX, D1=25.4*SY
      SX    = 0.3937 INCH = 10 MM
      SY    = 1.9685 INCH = 50 MM
      PLIST  = 0
                  0
                 16
                 22
                 20
                 44
                 28
```

Note that the first two values in the PLIST are not used (zeros were used as place holders).

The detail graphics produced in this example are shown below.



Notes

Consider the following when using COMPILE PROCEDURE:

1. Keywords consist of one to five characters.
2. A maximum of 29 parameters is represented by the variables T1, T2, T3, and so on. Substitution of other variable names is not allowed.
3. A maximum of fifteen keywords is allowed per line.
4. A maximum of 30 characters is allowed per expression (KEYWORD=DATA).
5. A maximum of approximately 1000 shapes (depending on the number of keywords used to define them) is allowed.

6. Mathematical operations within expressions are limited to the basic functions: addition (+), subtraction (y), multiplication (*), exponentiation (^), division (/), SQRT, SIN, COS, and TAN.
7. All parameters and constants (whether embedded in an expression or not) are interpreted in the unit declared with UNITS=. Component scales are in the component modeling unit. If a component is modeled in a unit different from that in the procedure file, the unit of the scales will differ from that of the constants, and parameters. To convert the scales, multiply the scale by a conversion factor. For example, L=SX*25.4 converts a scale in inches to millimeters.
8. The TOR shape does not take into account the long tangent on long-tangent elbows.
9. The position of the CLEARANCE keyword affects the interpretation of the SHAPE=definitions. If the CLEARANCE keyword is not entered, then the definition is part of the component description and is used for clearance checking. If the same definition is entered above and not below the CLEARANCE keyword, it is not used for clearance checking. If entered below and not above the CLEARANCE keyword, the definition is used for clearance checking and not for detail or surface generation.
10. The shapes GTRN, GRRT, and GRTR can be controlled in two ways:

If the shape is used as a nodal figure (or part of a nodal figure), as in a duct fitting, the information required to determine the position of the two faces with respect to one another can be determined from the nodal figure. The procedure file would contain, for example:

```
SHAPE=GTRN, W=T1, H=T2, W2=T3, H2=T4
```

The length of the shape and the position of faces would be read from the nodal line of the NFIG. If there is no NFIG, the software will assume that YOFF, ZOFF, and L are zero.

If the length and offset of the faces is not to be read from the nodal figure, or if no nodal figure is present, the information can be supplied directly via the procedure file, as in

```
SHAPE=GTRN, W=T1, H=T2, W2=T3, H2=T4, L=T5,YOFF=T6, ZOFF=T7
```

or

```
SHAPE=GTRN, W=T1, H=T2, W2=T3, H2=T4, L=20,YOFF=30, ZOFF=40
```

Either of the above two cases causes information that could be read off of a nodal line to be ignored.

CREATE RENDERING

Description

This command turns nodal figures, nodal lines with DUCTSIZE or OUTERDIAM properties, or nodal figures that use strings representing planar faces of objects (such as projected outlines, or projected polygons) into three-dimensional shapes with hidden line removal. It also erases all appropriate lines when a shape is in front of another shape in order to create realistic three-dimensional drawings.

Hidden lines can be shown symbolically. The DASH modifier displays hidden lines as dashed lines rather than erased lines.

The nodal figures and nodal lines are shape representations of what are to become three-dimensional designs. Nfigures must be created and have procedure files to define their component shapes. Nlines with the DUCTSIZE or OUTERDIAM properties (inserted with the commands ROUTE DUCT (see the *Heating, Ventilating, and Air Conditioning Reference*) or ROUTE PIPE and nodal figures that use strings representing planar faces of objects inserted with the commands PROJECT OUTLINE (see the *Visualization/Preparation Reference*), and PROJECT POLYGON (see the *Area Information Management Reference*) do not need procedure files for hidden line removal.

Entities made from one or more basic shapes (e.g., box, cylinder) must be prepared as nodal figures and must be defined by a related procedure file. For example, a three-dimensional rectangular table with a square base and a round pedestal is created with three shapes: a rectangular box, a square box, and a cylinder. You can create the associated procedure file to define the relationship of these shapes by using the command CONSTRUCT PROCEDURE (see the *Visualization/Preparation Reference*).

Please note: It is not necessary to compile procedure files used by CREATE RENDERING.

Valid Modes

Model mode and Draw View mode. Activate Model mode with this command:

```
#n# SELECT MODE MODEL
```

Activate Draw View mode with this command:

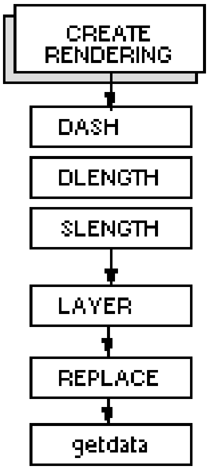
```
#n# SELECT MODE DRAW VIEW: view d1
```

Syntax

```
#n# CREATE RENDERING [modifiers]: getdata
```

Modifiers

The following diagram shows the flow of CREATE RENDERING modifiers. For a full explanation of modifier diagrams, see Notes to Readers.



Modifiers are listed alphabetically by level. When there are multiple levels of modifiers, level is indicated by the left number. A description of each modifier follows.

DASH: Displays hidden lines as dashed lines when these lines are on a face that is partially invisible. However, when entire faces of rendered graphics are invisible, no dashes are generated. The standard line font DASHED is used.

DLENGTH x: Dash Length. Specifies the length of a dash as a positive real number in drawing units. The range is 0.0 to 32,767.0. The default is 1.0.

SLENGTH x: Space Length. Specifies the length of blank segments between dashes as a positive real number in drawing units. The default is 1.0.

LAYER n: Inserts rendered graphics on layer n. You can use 0 to 254. The default is the active layer. Projected outlines are rendered on the layer they were projected onto even if this modifier is used.

REPLACE: Creates new renderings for old rendered shapes. Selected entities are considered in relation to each other. A hidden portion of a selected shape is not erased unless all overlapping shapes are also selected.

The default is to ignore existing rendered shapes.

Please note: Before using the command with the REPLACE modifier, you must execute SELECT RELATIONSHIP ON to relate original nodal figures to their renderings.

Procedure

```
#n# SELECT RELATIONSHIP ON
PARENT IDENT FLAG ON
#n# CREATE RENDERING [modifiers]: MODEL ent d1... dn
```

1. Enter SELECT RELATIONSHIP ON if some portion of the model has already been rendered to relate original nodal figures to their renderings.
2. Press RETURN. The system prompts `PARENT IDENT FLAG ON.`
3. Enter CREATE RENDERING.
4. Enter any modifiers you want to use.
5. Enter a colon.
6. If no view was defined by SELECT MODE DRAW VIEW, after the `view` prompt, select the desired view for the rendering.
7. After the `MODEL ent` prompt, select “three-dimensional” nodal figures to erase all hidden lines.
8. Press RETURN.

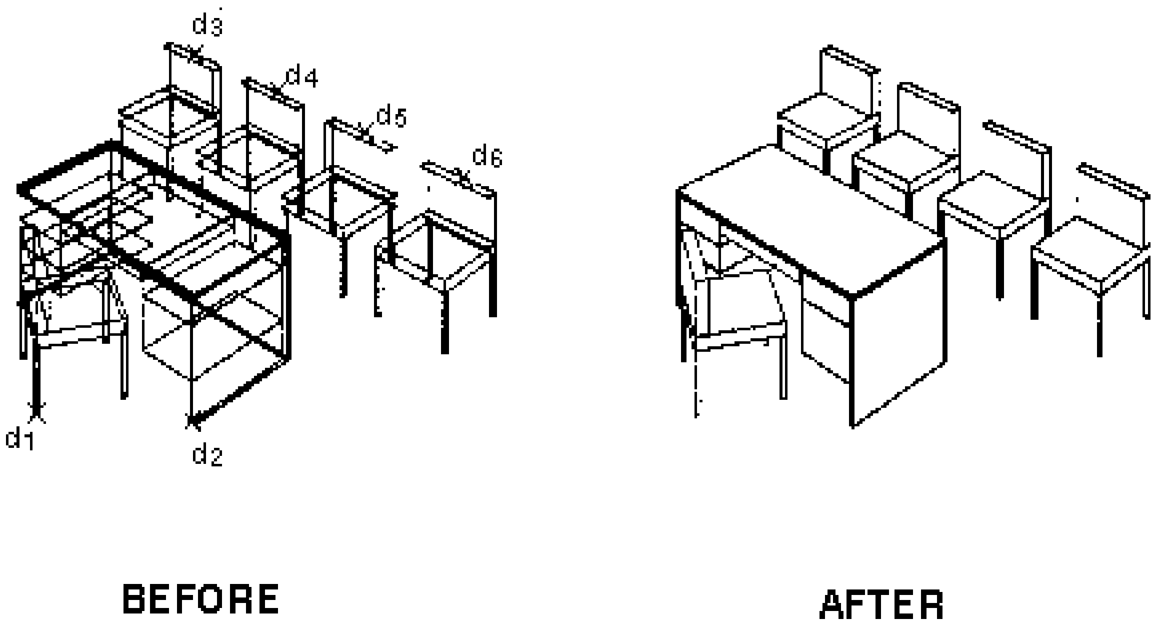
The selected nodal figures become realistic-looking, three-dimensional appearances with all hidden lines erased.

Example 1

In this example, a desk and a chair are shown as they were created with the CONSTRUCT PROCEDURE command. Then, they were inserted into a part and CREATE RENDERING was used to create the final rendering.
CREATE RENDERING erases all hidden lines, creating a realistic-looking, three-dimensional representation of the shapes.

```
#n# CREATE RENDERING: MODEL ent d1d2d3d4d5d6
```

```
6 COMPONENTS BEING PREPARED.PROCESS COMPLETED, GENERATING
GRAPHICS.
```



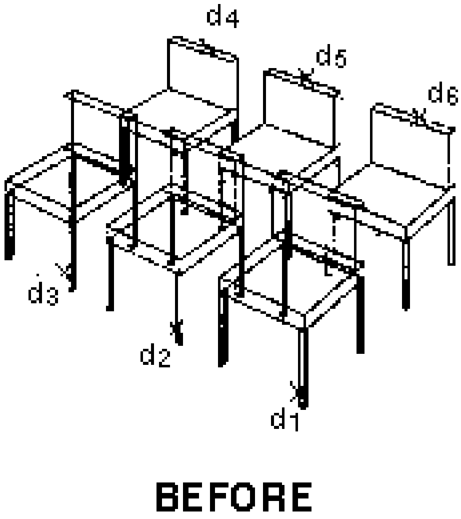
Example 2

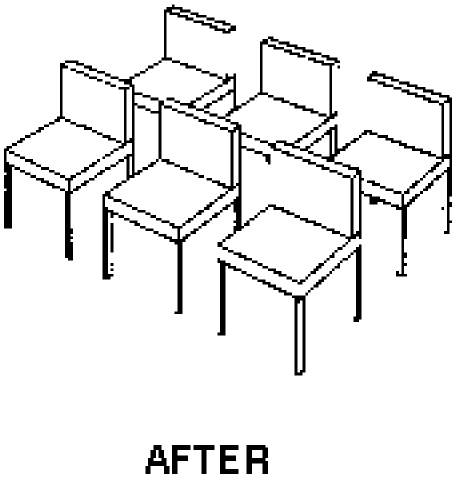
This example uses the REPLACE modifier. First, three chair Nfigures were inserted and rendered. Second, three additional chairs are inserted, partially overlapping the chairs in the back row. Third, CREATE RENDERING REPLACE is used to erase (1) hidden lines of the chairs in front so that each looks three-dimensional and (2) hidden portions of the chairs in back.

If the REPLACE was not used, the chairs in the back row, already rendered, would be ignored; their hidden portions would not be erased.

```
#n# SELECT RELATIONSHIP ON
PARENT IDENT FLAG ON
#n# CREATE RENDERING REPLACE: MODEL ent 1d2d3d4d5d6d
```

```
6 COMPONENTS BEING PREPARED.PROCESS COMPLETED, GENERATING
GRAPHICS.
```





Notes

Consider the following when using CREATE RENDERING:

1. Projected objects cannot intersect each other in model space unless a hole has been previously defined.
2. All strings and arcs must be planar if they are to be hidden correctly. Violation of this restriction may result in unpredictable renderings.
3. After performing any dynamic operation (DYNAMIC VIEW) on a view, use the SET VIEW or RESET VIEW command. This prevents subsequent renderings from being lost from a temporary view when you file the part.
4. Rendered shapes only simulate a solid, three-dimensional appearance. Because they are for appearance only, rendered graphics cannot be used in getdata with point-on-entity techniques.
5. To identify individual strings and arcs of rendered shapes when you are executing other commands, first use:

```
#n# SELECT RELATIONSHIP OFF
#n# SELECT MODE DRAW VIEW: view d1
```

Then, when a command prompts `DRAW ent`, select individual strings and arcs. You can use the getdata modifiers STR and ARC.

To identify entire rendered shapes (made from nodal figures) when you are executing other commands, first use SELECT RELATIONSHIP ON in Model mode to relate rendered shapes to their nodal figures. When you are prompted `MODEL ent`, select anywhere on the rendered nodal figure. You can use the getdata modifier NFIG.

6. Rendered shapes do not survive operations on the related nodal figure if the operation invalidates their appearance. For example, if you translate, delete, or rotate a nodal figure or translate, delete, rotate, stretch, or break a nodal line, its rendered graphics are deleted.
7. When shapes overlap, the command erases all hidden lines only if these shapes are rendered within a single execution of the command.

For example, if you render a chair with one execution of CREATE RENDERING, insert another chair partially behind the first, and then use the command again to remove the hidden lines of the second chair—only the hidden lines that actually make up the second chair are erased. The hidden portion of the background chair is not processed.

Use the MERGE RENDERING command to hide strings or arcs of overlapping nodal figures that were rendered with separate executions of CREATE RENDERING.

8. For each execution of the command, erasing is done only for newly inserted Nfigures. Rendered shapes from previous executions of CREATE RENDERING and MERGE RENDERING remain unchanged unless you use REPLACE.
9. When nested nodal figures are involved, identify each nodal figure individually.
10. You can render the same nodal figure in multiple views, but in only one view at a time.
11. There are two ways to regenerate original nodal figures:

Use the DELETE VIEW command and then define a new view with the identical name and parameters.

Delete all the associated rendered entities in a rendered view. Use the commands SELECT MODE DRAW VIEW and DELETE ENTITY.

12. In addition to generating rendered shapes automatically, you can add to rendered shapes interactively. First, use the command SELECT MODE DRAW VIEW. Second, use INSERT STRING and INSERT ARC. Third, use the RELATE DETAIL command to relate the rendered entities to the original nodal figures.

13. ESC-Q aborts the command only if you use it before this message appears: `n COMPONENTS BEING PREPARED`. The message indicates that the rendered shapes have been generated and automatic hiding is about to start.

After this point, the hiding process is completed even if you use ESC-Q. The rendered shapes generated before you use ESC-Q are not automatically displayed. To display them, use the command REGENERATE GRAPHICS. Use MERGE RENDERING or EDIT APPEARANCE to do any corrective line hiding.

14. Entities of these nodal figures are specific to the selected drawing and view. Rendered shapes have coordinates in model rather than drawing space; however, these graphics are not part of the model (as is the original nodal figure). They cannot be used in getdata with point-on-entity techniques.

EDIT APPEARANCE

Description

This command erases, dashes, or restores a specified portion of the graphic representation of identified entity(s) within a specified view. The specified portion may be a window, polygon window, or a line segment. Selecting one string of a projected wall, projected outline, or displayed beam identifies all the strings for that entity.

This command operates on the arcs, lines, and strings in an Nfigure but not on the Nfigure itself. This means, for example, that appearance symbols and related Ntext are not blanked or dashed.

To restore items that have been completely blanked with this command, select them in another view and define the region for restoration in the view where you want them unblanked.

Please note: Do not use this command in perspective views.

Valid Modes

Model or Draw mode with a view selected.

To enter Draw mode, use the following command:

```
#n# SELECT MODE DRAW VIEW: view d1
```

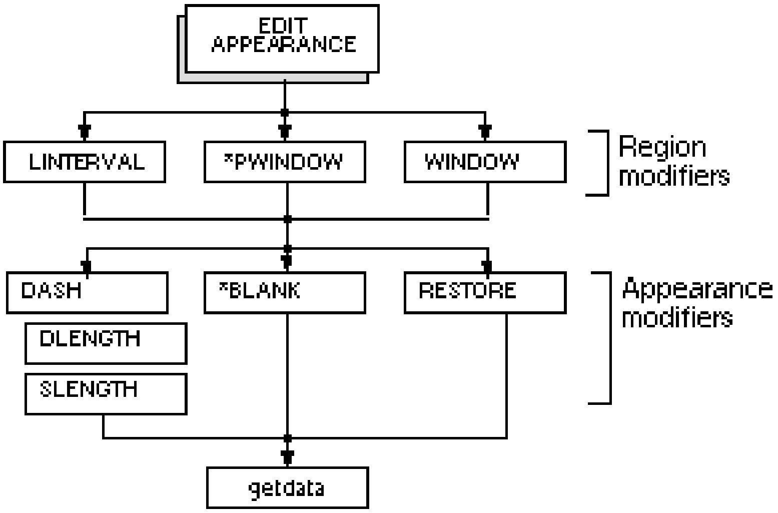
Syntax

```
#n# EDIT APPEARANCE region modifier appearance modifier
[modifiers]: getdata
```

Select both a Region modifier, to define the region for processing, and an Appearance modifier, to specify blanking, dashing, or restoring. If you do not select a Region modifier, PWINDOW is the default. If you do not select an Appearance modifier, BLANK is the default.

Modifiers

The following diagram shows the flow of EDIT APPEARANCE modifiers. An asterisk (*) denotes a default modifier. For a full explanation of modifier diagrams, see Notes to Readers.



Modifiers are listed alphabetically by level. When there are multiple levels of modifiers, level is indicated by the left number. A description of each modifier follows.

1 BLANK: (Default) Blanks portions of identified entities in the specified region of the specified view.

1 DASH: Dashes portions of identified entities in the specified region of the specified view.

2 DLENGTH x: Dash Length. Specifies the length of the dash. The default is.1 model units.

2 SLENGTH **x**: Space Length. Specifies the length of the blank segments between dashes. The default is.1 model units.

1 LINTERVAL: Linear Interval. Limits processing to the line segment between two points specified in getdata.

1 PWINDOW: (Default) Limits processing to a region defined by a polygon specified by three to twenty points in getdata.

A maximum of 1000 entities can be identified within a PWINDOW; a maximum of 15,000 entities related to these entities can be processed.

1 RESTORE: Restores portions of identified entities in the specified region of the specified view.

1 WINDOW: Limits processing to a region defined by a rectangle specified by two points (corners) in getdata.

A maximum of 1000 entities can be identified within a WINDOW; a maximum of 15,000 entities related to these entities can be processed.

Procedure 1: Using the WINDOW and PWINDOW Modifiers

```
#n# EDIT APPEARANCE <WINDOW or PWINDOW> <BLANK, DASH, or RESTORE>
[modifiers]: view d1 MODEL loc d1...dn; MODEL
ent dn+1...dn +m
```

1. Enter EDIT APPEARANCE.
2. Enter either WINDOW or PWINDOW (to define the region for blanking, dashing, or restoring) and BLANK, DASH, or RESTORE (to specify the action).
3. Enter additional modifiers as desired.
4. Enter a colon. In response to the `view` prompt (which only appears in Model mode), select the desired view.
5. In response to the `MODEL loc` prompt, select either two points (for the WINDOW) or from three to twenty points (for the PWINDOW).
6. Enter a semicolon. In response to the `MODEL ent` prompt, select the entities you want to blank, dash, or restore. All Nfigures related to the selected entity are identified.
7. Press RETURN.

Procedure 2: Using the LINTERVAL Modifier

```
#n# EDIT APPEARANCE LINTERVAL BLANK, RESTORE, or DASH [modifiers]:
view d1 MODEL ent d2 MODEL ent d3
```

1. Enter EDIT APPEARANCE.
2. Enter LINTERVAL to specify the region for blanking, dashing, or restoring and BLANK, RESTORE, or DASH to specify the action.
3. Enter additional modifiers as desired.
4. Enter a colon. In response to the `view` prompt (which only appears in Model mode), select the desired view.
5. In response to the first `MODEL ent` prompt, select the line where you want the segment to start. In response to the second `MODEL ent` prompt, select the end of the segment.

Note that the line segment cannot contain a vertex but that you can specify a vertex as the end of the segment.

6. Press RETURN.

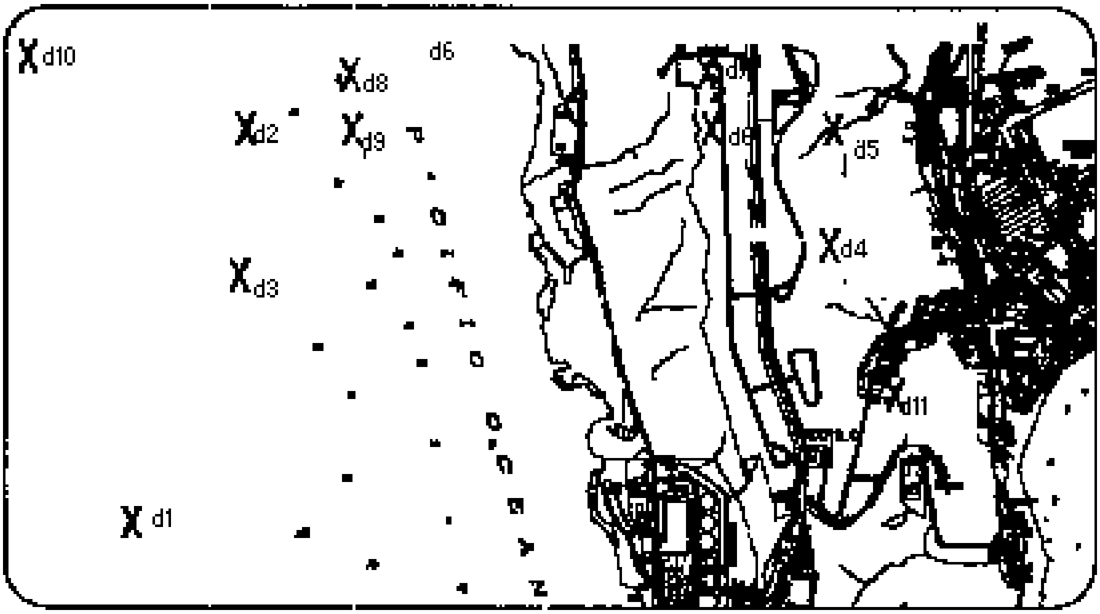
Example

In this example, the modifier PWINDOW is used to blank a portion of a map. This command produces the following illustrations:

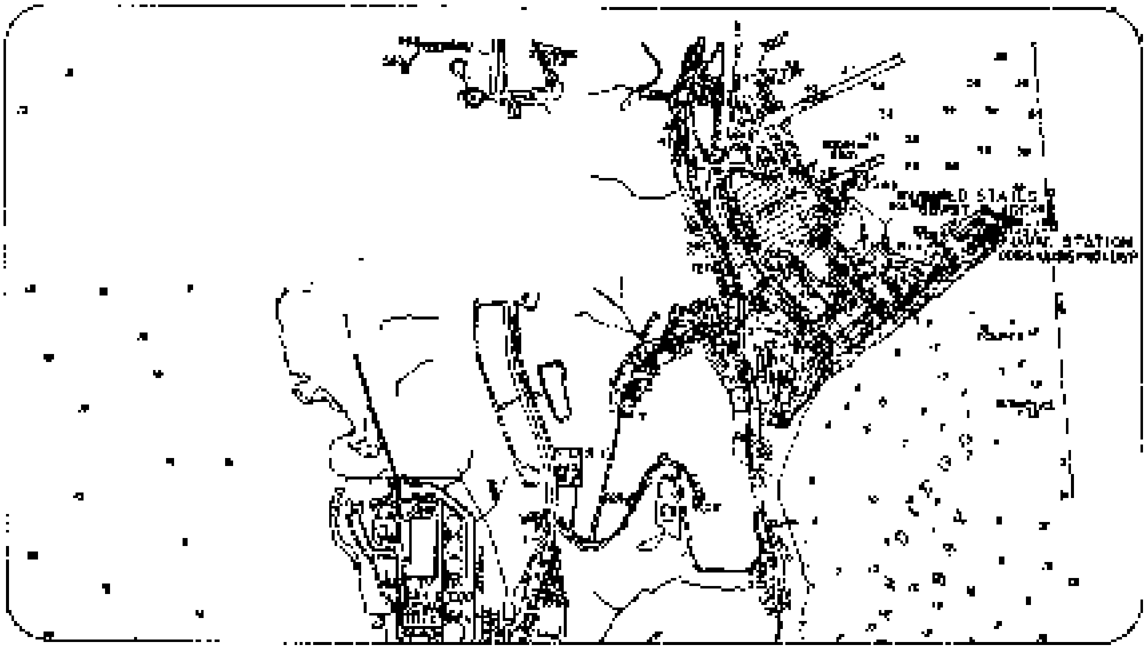
```
#n# EDIT APPEARANCE PWINDOW: view d1; MODEL loc d2d3d4d5d6d7d8d9
MODEL ent WIN IN CROSS d10 d11
```

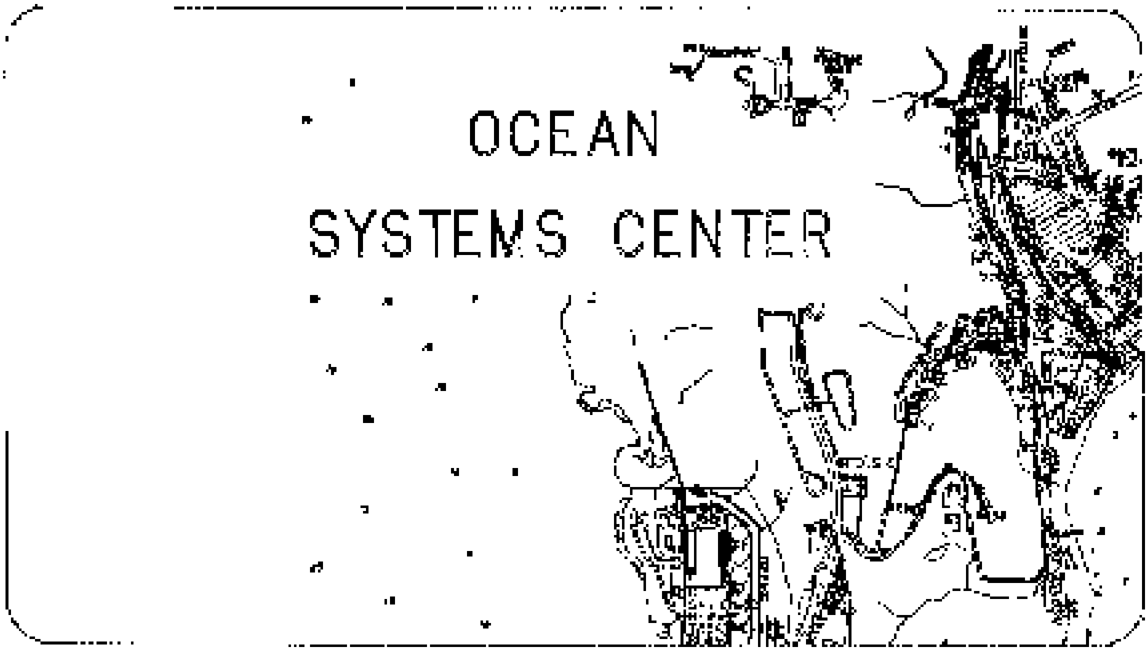
After the view is selected (d1), the polygon region for blanking is specified (d2...d9). A window is used to identify the entities to be blanked (d10 d11).

The first AFTER illustration shows the map with the blanked area; the second shows the map with sample text inserted.



BEFORE





AFTER

GENERATE DETAIL

Description

This command generates three-dimensional appearances and clearance representations for selected items. Piping is represented with either single or double lines. Other appearance representations (e.g., ducting, fittings, equipment) and clearance representations are made from shapes such as cylinders, cones, and boxes. Appearance representations are enhanced by hidden line removal.

Procedure files define the shapes that generate the representations. PNAME, DUCTSIZE, or OUTERDIAM properties specify which procedure is to be used, with DUCTSIZE taking precedence over the other two. The PNAME property on a nodal line (Nline) or nodal figure (Nfigure) points to a procedure file. If it is not present on the Nline or Nfigure, a procedure file with the same name as the Nfigure or one associated with a parent Nfigure is used. DUCTSIZE indicates a cylinder (1 size) or a box (2 sizes) the length of the nodal line. OUTERDIAM indicates a cylinder the length of the Nline. Parameters for other procedures are taken from the PLIST property on the Nline or Nfigure or from the default parameter file.

When you use the command GENERATE DETAIL, any uncompiled procedure is compiled automatically.

GENERATE DETAIL can be used in either Draw view or Model mode. Note that in Model mode the modifier REPLACE automatically invokes SELECT RELATIONSHIP ON. When completed, the system returns to SELECT RELATIONSHIP OFF. The created graphics are drawing entities and are view specific; they are children of the model (parent) entity they detail. You can identify an item in a view other than the one you want detailed as long as it is not the last item identified in the command; the graphics are generated in the last view that is selected.

Although the created graphics are drawing entities and are not part of the model, they have their coordinates in model rather than drawing space. They may not be used in getdata with point-on-entity techniques to reference model geometry. However, the model is referenced through the graphics when you use the modifier REPLACE.

GENERATE DETAIL erases the model graphics (except Nlines and Cnodes within Nfigures) in the specified view and replaces them with detail graphics. The process automatically hides the appropriate lines in items identified within the command. HIDE PIPE and UNHIDE PIPE edit hidden lines in items detailed in separate executions of the GENERATE DETAIL command.

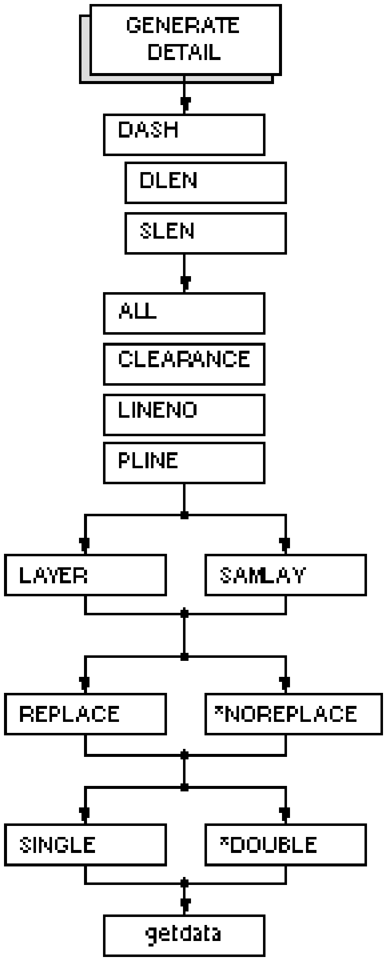
Valid Modes: Model or Draw view mode

Syntax

```
#n# GENERATE DETAIL <LINENO or ALL> [modifiers]
#n# GENERATE DETAIL [modifiers]: <getdata>
```

Modifiers

The following diagram shows the flow of GENERATE DETAIL modifiers. An asterisk (*) denotes a default modifier. For a full explanation of modifier diagrams, see Notes to Readers.



Modifiers are listed alphabetically by level. When there are multiple levels of modifiers, level is indicated by the left number. A description of each modifier follows.

1 ALL: Generates representations for all entities. If more than one view exists in the drawing and no view has been selected, you are prompted to select the view you want detailed.

1 CLEARANCE: Generates clearance representations for selected entities. If items already have appearance representations (from a previous use of the command), you must use the REPLACE modifier to replace them with clearance graphics.

1 DASH: Displays hidden lines as dashed lines instead of blanking them. The default font (DASHED) is used. Note that lines hidden due to the component being hidden against itself are also dashed. The DASH modifier should not be used with a large number of entities since interactive touch-up may be required.

2 DLEN x: Dash Length. Specifies dashing and the length of a dash in drawing units. The default is 0.16 inches regardless of drawing units. If a drawing is in millimeters, the default is listed as 4.064.

2 SLEN n: Space Length. Specifies dashing and the length of blank segments between dashes in drawing units. The default is 0.16 inches regardless of drawing units. If a drawing is in millimeters, the default is listed as 4.064.

1 DOUBLE: (Default) Details all piping with double line graphics; deactivates the modifier SINGLE.

1 LAYER n: Inserts detail graphics on layer n, between 0 and 254. If n is negative, the command uses 0 unless n = y1. In the latter case, the active construction layer is used. Any number greater than 254 defaults to 254.

The default is the active construction layer. The layer specified in SELECT PPARAMETERS does not apply to detailing, only to modeling.

1 LINENO text: Line Number. Details a pipeline with the specified line number. The line number may include any adjacent fields in the pipeline’s label, for example, CW-100. Only the first pipeline found with the specified line number is identified. No getdata is required or allowed.

1 NOREPLACE: (Default) Generates appearance or clearance representations for selected items that have not been previously detailed.

1 PLINE: Pipeline. Details every pipe and fitting in identified pipelines. Stand-alone components, such as pieces of equipment and structures can still be selected. Each selection on a pipeline component identifies all the components in the pipeline.

1 REPLACE: Generates appearance or clearance representations for selected items, replacing any detail graphics that were generated in a previous use of the command.

1 SAMLAY: Same Layer. Creates detail entities on the same layer as the corresponding model entity. Without SAMLAY detail entities go to the active construction layer or to the layer specified with the modifier LAYER.

1 SINGLE: Details piping with either single- or double-line graphics depending on the diameter specified with SELECT PPARAMATERS ABOVE. Piping with a diameter above the size specified becomes double-line. Piping with a diameter equal to or below the size specified becomes single-line. Before using SINGLE, you must set the size for ABOVE with SELECT PPARAMETERS (see *Piping Reference*).

Procedure 1

```
#n# GENERATE DETAIL [modifiers]: MODEL ent d1...dn
```

1. Enter GENERATE DETAIL and any modifiers you want to use.
2. After typing a colon, respond to the prompt `MODEL ent` by selecting individual pipes (Nlines) or other items (Nfigures).
3. With the PLINE modifier, select a pipe on each of the pipelines you want to detail.
4. Press RETURN.

Procedure 2

```
#n# GENERATE DETAIL LINENO line number [modifiers]
```

1. Enter GENERATE DETAIL LINENO and the line number of the pipe to be detailed.
2. Enter any additional modifiers you want to use.
3. Press RETURN.

Procedure 3

```
#n# GENERATE DETAIL: MODEL ent d1...dn
```

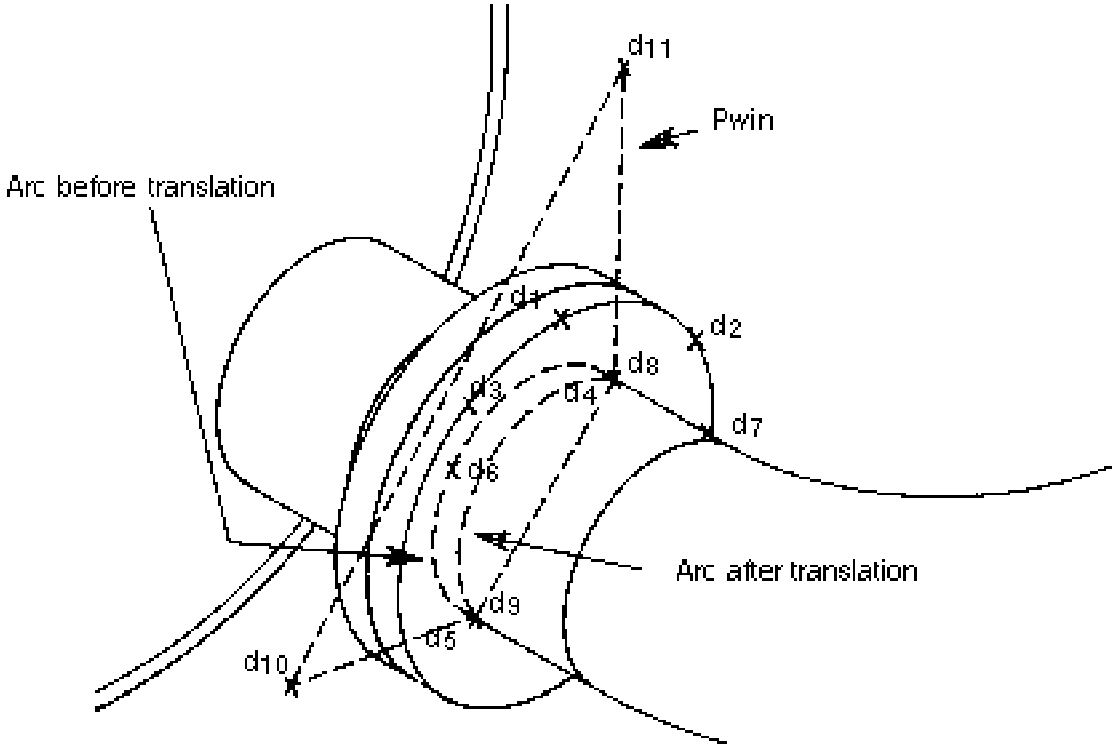
1. Enter GENERATE DETAIL and any other modifiers you want to use.
2. In response to the prompt `MODEL ent`, select the detail graphics to identify the model entities. (It is sufficient to select once per model entity, anywhere on its detail graphics.) You can use the modifiers NLINE and NFIG in getdata to filter the model entity selection. Note that you can also select nonerased model graphics within the same command.
3. Press RETURN.

Example 1

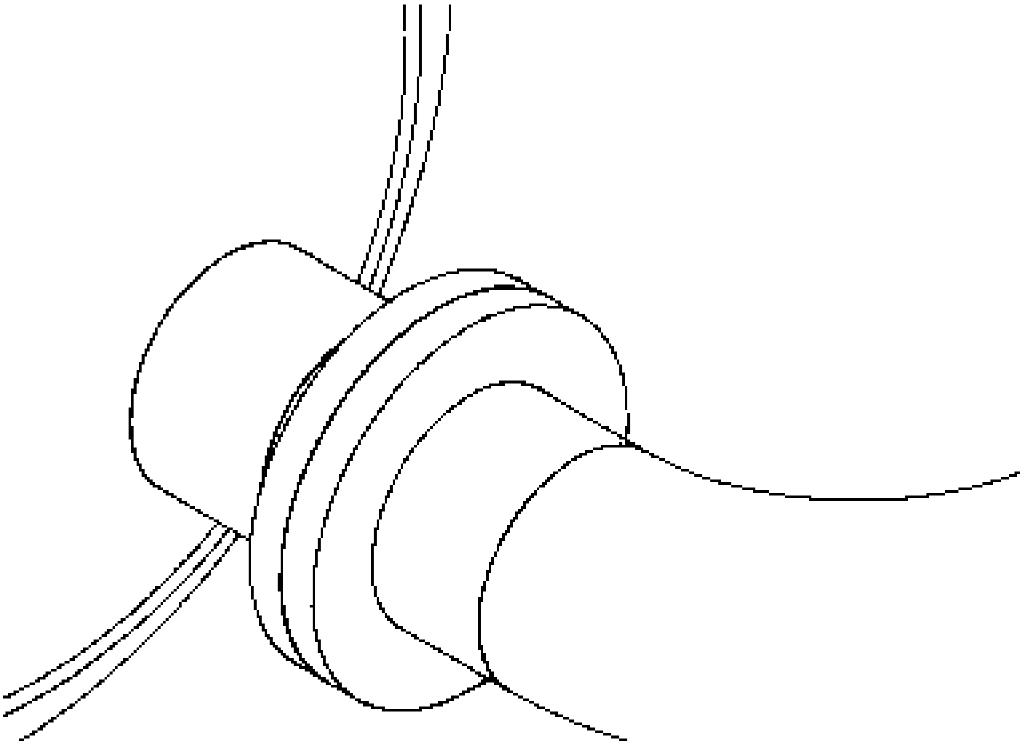
The first example shows the detailing of a few items that are individually identified with a window. First the parameter file for the CVPDC model is activated by turning on the directory file, which specifies the parameter files that go with different layer ranges. After the detailing is done, the nodal entities are echoed off in the detailed view.

```
#n# SELECT PPARAMETERS DRFILE CVPDC.DRFILE
#n# GENERATE DETAIL: MODEL ent WIN MODEL loc IN CROSS d1d2
6 COMPONENTS BEING SOLVED
. . . . . . . . . . .
. . . . . . . . . . .
(regenerating graphics)
```

```
#n# ECHO ENTITY CNODE TNODE NLINE NTEXT OFF: view d3
```



BEFORE



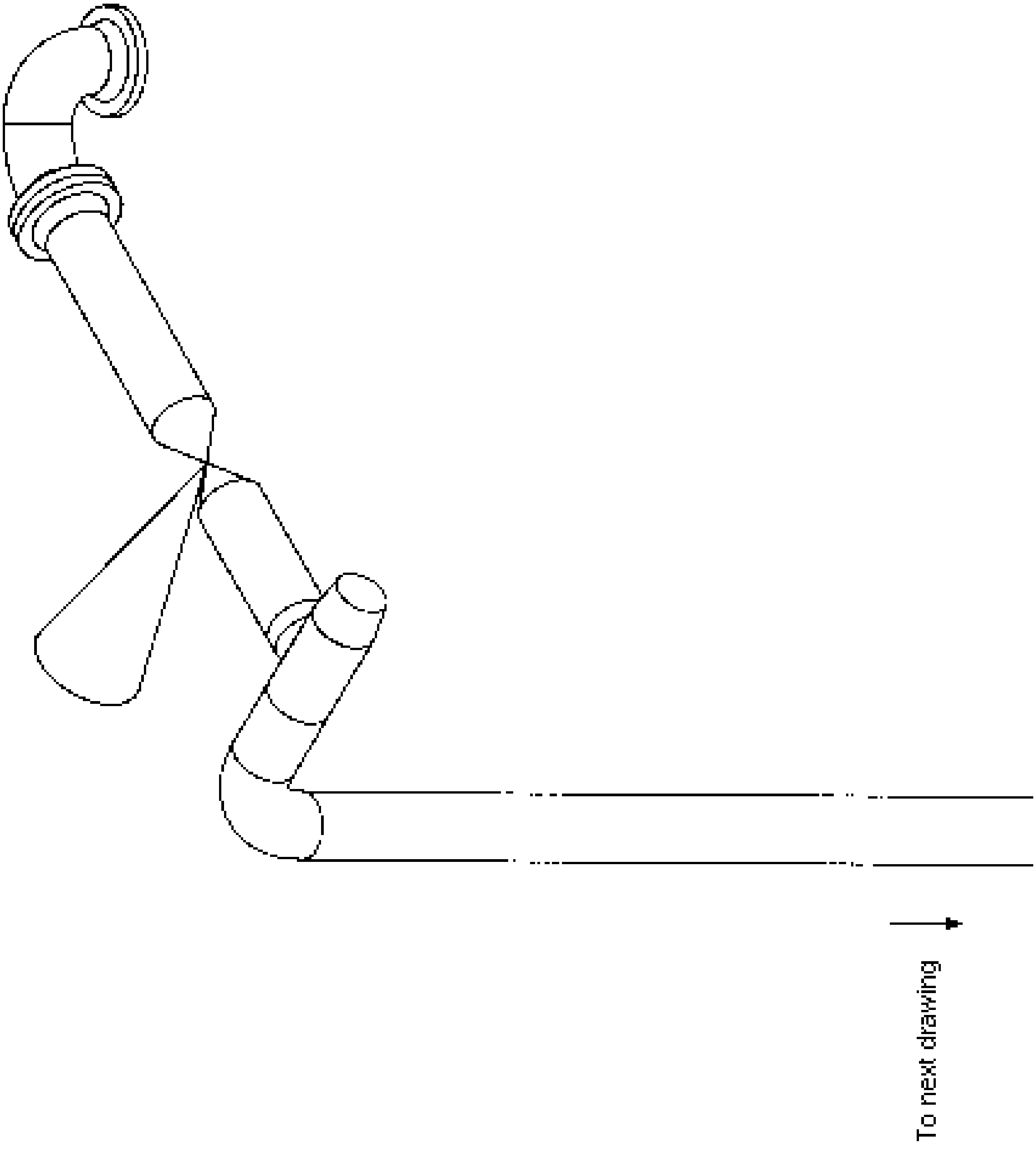
AFTER

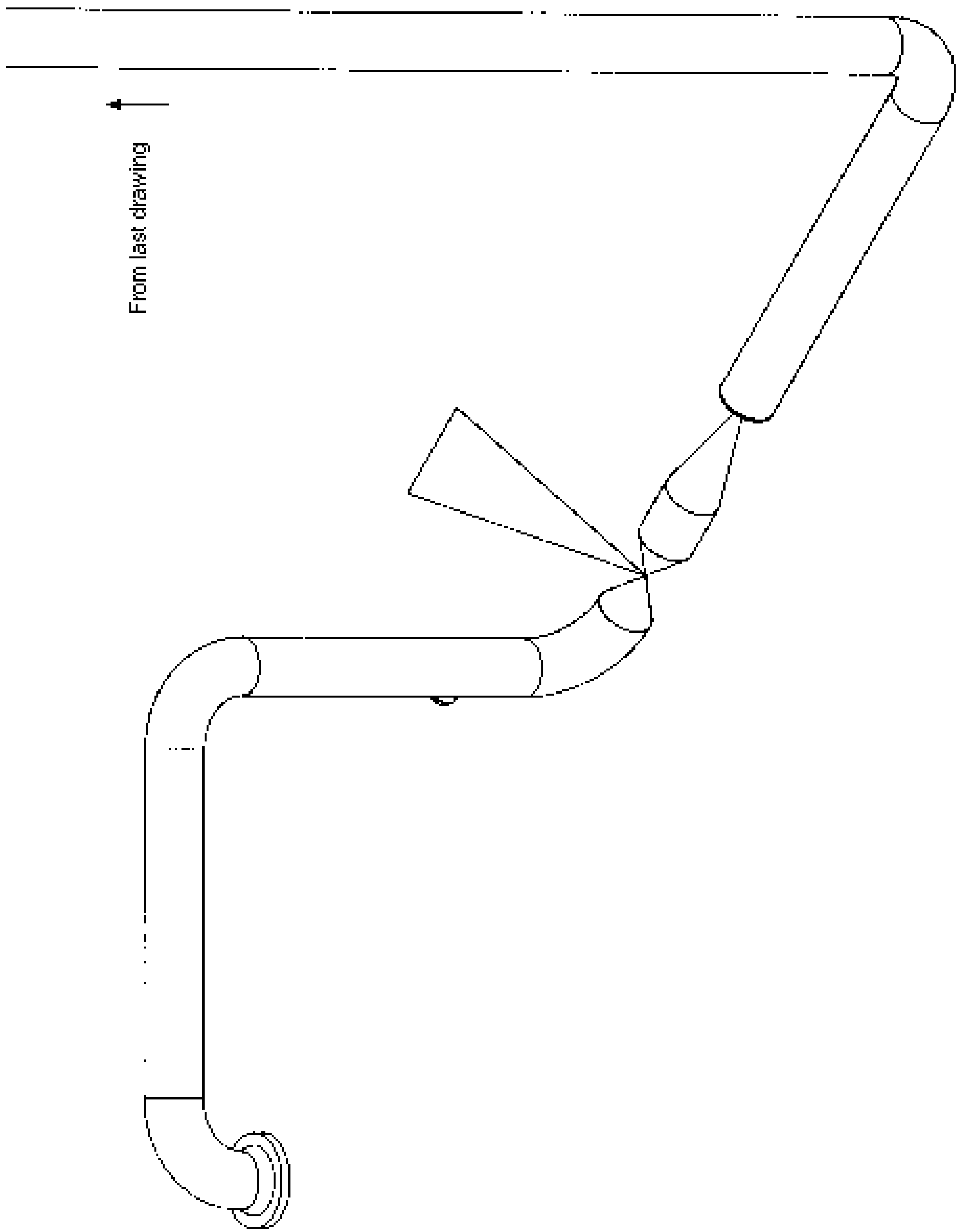
Example 2

In this example, the LINENO modifier is used to detail a whole pipeline identified by its line number. No getdata is needed. A pipeline can also be identified by a single selection anywhere along the line when you use the PLINE modifier and enter getdata with a colon.

```
#n# GENERATE DETAIL LINENO O-8242
36 COMPONENTS BEING SOLVED
```

```
.  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .
(regenerating graphics)
```




From last drawing

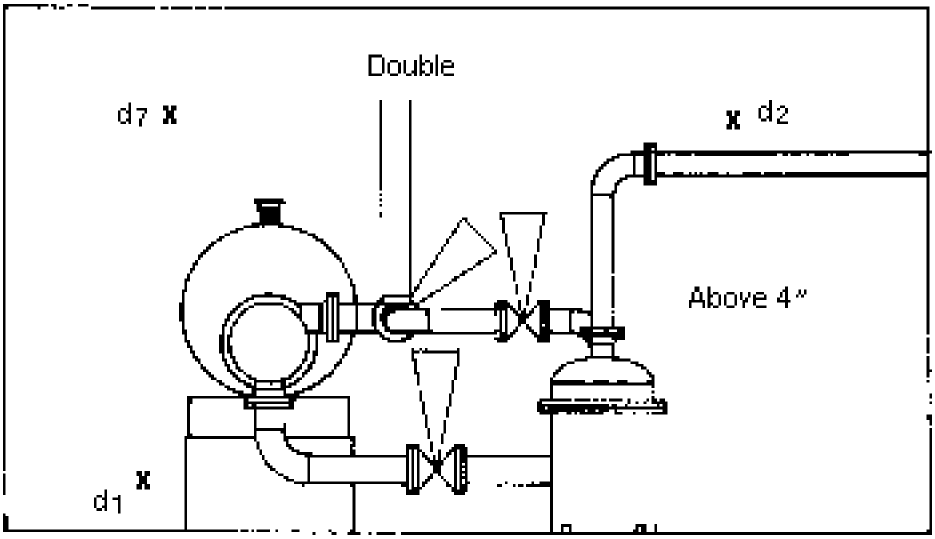
Example 3

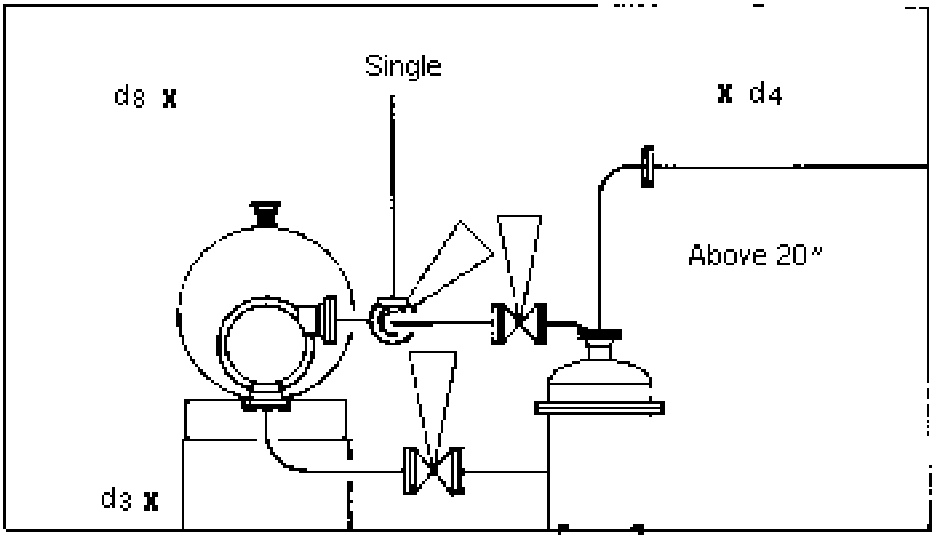
Double line graphics are generated by default. This three-part example shows how to combine the SINGLE modifier with the GENERATE DETAIL command and the ABOVE modifier with the SELECT PPARAMETERS command to generate single and double line details.

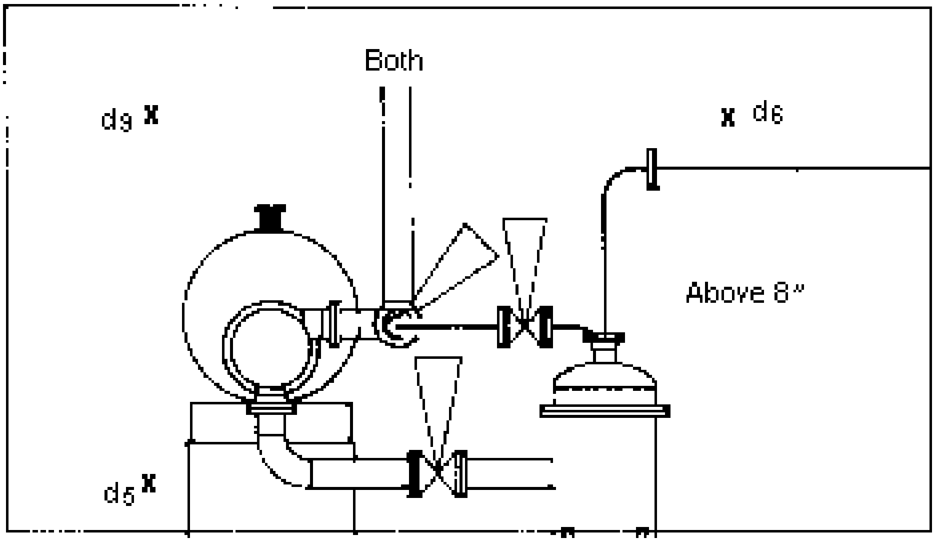
The first part of the example (DOUBLE in the illustration) specifies double-line graphics for all pipes ABOVE four inches; this results in all double-line graphics since there are no pipes of less than four inches. In the second part (SINGLE in the illustration), the ABOVE 20 inches results in all single-line graphics. In the BOTH part of the example, the ABOVE 8 inches results in some single- and some double-line graphics.

Use the command INSERT PBREAK to produce single- or double-line pipe break symbols (see the *Piping Reference* for more information).

```
#n# SELECT PPARAMETERS PIPE ABOVE 4″
#n# GENERATE DETAIL SINGLE: MODEL ent WIN MODEL loc IN CROSS d1 d2
52 COMPONENTS BEING SOLVED
. . . . . . . . . . .
. . . . . . . . . . .
(regenerating graphics)
#n# SELECT PPARAMETERS PIPE ABOVE 20″
#n# GENERATE DETAIL SINGLE: MODEL ent WIN MODEL loc IN CROSS d3d4
52 COMPONENTS BEING SOLVED
. . . . . . . . . . .
. . . . . . . . . . .
(regenerating graphics)
#n# SELECT PPARAMETERS PIPE ABOVE 8″
#n# GENERATE DETAIL SINGLE: MODEL ent WIN MODEL loc IN CROSS d5d6
52 COMPONENTS BEING SOLVED
. . . . . . . . . . .
. . . . . . . . . . .
(regenerating graphics)
#n# ECHO ENTITY NLINE OFF: view d7d8d9
```


Double
d7 x
x d2
Above 4"
d1 x

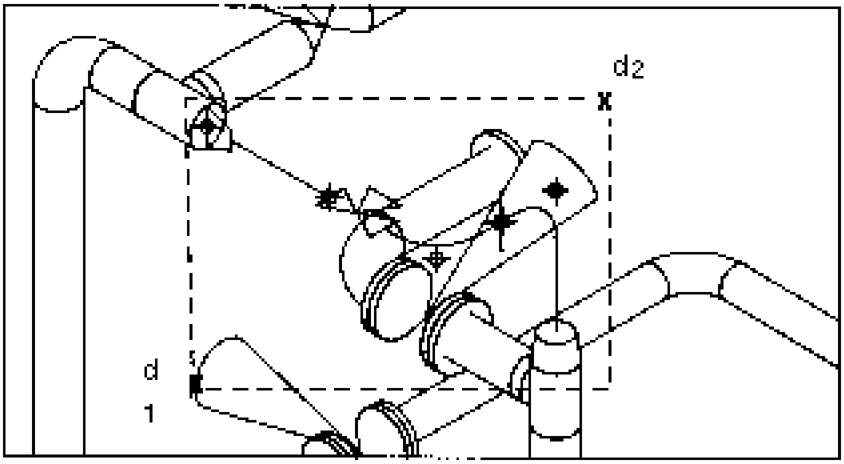

Single
d8 x
x d4
Above 20"
d3 x


Both
d9 x
x d6
Above 8"
d5 x

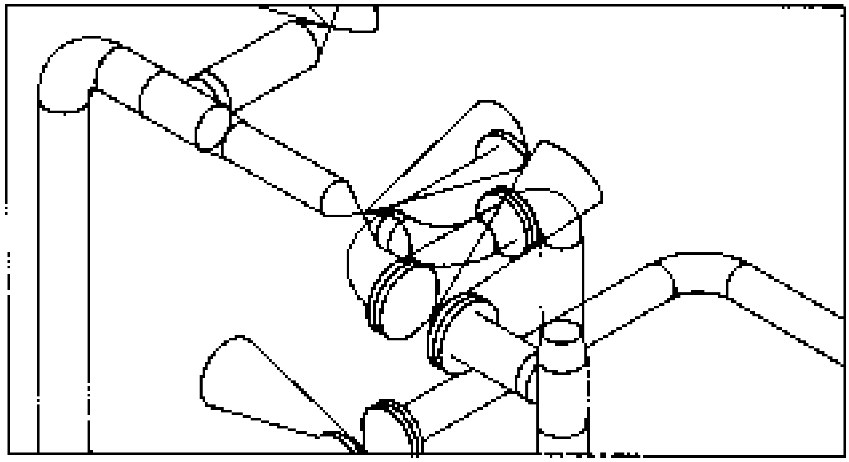
Example 4

This example shows how to detail entities added to a drawing with or without replacing the details of existing entities. The REPLACE and NOREPLACE options are compared. Note that hiding between the pipe and flange in the circled area was undone by REPLACE. Use the HIDE PIPE command to redo the hiding between these two items as shown in Example 1 in the description of HIDE PIPE.

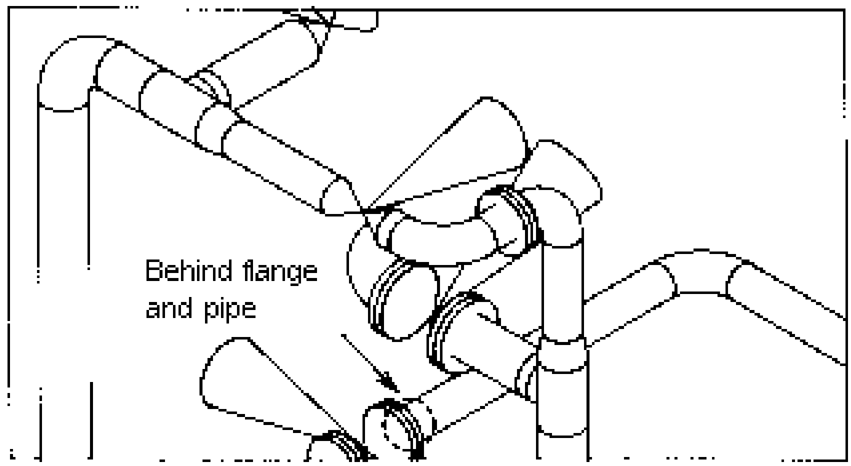
```
#n#SELECT RELATIONSHIP ON
#n#SELECT MODE DRAW VIEW:view NAME ISO2
#n#GENERATE DETAIL NOREPLACE:MODEL ent WIN DRAW loc IN CROSS d1d2
11 COMPONENTS BEING SOLVED
. . . . . . . . . . .
. . . . . . . . . . .
(regenerating graphics)
#n#SELECT MODE DRAW VIEW:view NAME ISO3
#n#GENERATE DETAIL REPLACE:MODEL ent WIN DRAW loc IN CROSS d1d2
23 COMPONENTS BEING SOLVED
. . . . . . . . . . .
. . . . . . . . . . .
(regenerating graphics)
#n# ECHO ENTITY NLINE OFF:view NAME ISO2 NAME ISO3
```



BEFORE



AFTER NOREPLACE



AFTER REPLACE

Note: Consider the following when using the GENERATE DETAIL command:

1. The command GENERATE DETAIL applies to Nfigures (nodal figures) and to duct or pipe Nlines (nodal lines). It can also be used on any Nline associated with a procedure file and with a list of parameters. You do not need to use an Nfigure because you can put the PNAME and PLIST properties directly on the Nline. This means that you can design complex pipelines, ducting, and structures using the primitive shapes that follow the path of the Nline. If the PNAME and PLIST properties are not present on an Nline or Nfigure, the command goes to an associated parent Nfigure to find the necessary appearance or clearance generation procedure.

Pipes and round ducts do not require procedure files for detailing because they have only one shape (a cylinder). The dimensions of the shape are defined by the length of the Nline and by the OUTERDIAM or DUCTSIZE property on the entity.

The parameters required by the procedure are retrieved either from a PLIST property that you insert onto the component or from a parameter file. In the latter case, GENERATE DETAIL uses stock numbers to find the component's parameters in the parameter file assigned to the layer on which the component resides. Parameter file to layer range assignments are made in your directory file.

2. After performing any dynamic operation (DYN VIEW) on a view, issue the SET VIEW or RESET VIEW command. This prevents subsequent detailing from being lost from a temporary view when you file the part.
3. To identify the entities created to form detail graphics, use these commands:

SELECT RELATIONSHIP OFF
SELECT MODE DRAW VIEW: d

Then when any command prompts `DRAW ent`, select the individual arc or string. You may use getdata modifiers ARC or STR to filter the entity selection.

To identify detailed model entities, use SELECT RELATIONSHIP ON in Model mode. The only exception to this occurs in the commands HIDE and UNHIDE PIPE. These require that you activate Draw mode with a selected view. In either of these cases, however, you must use SELECT RELATIONSHIP ON to identify detailed model entities. When you are prompted `MODEL ent`, select anywhere on the detail graphics related to the model entity. You may use getdata modifiers NLIN or NFIG to filter the entity.

4. Detail graphics do not survive any operation on the related model entity if that operation invalidates their appearance. For example, if you translate a model

entity, its detail graphics go with it; however, if you stretch, delete, break, or rotate a model entity, its detail graphics are deleted.

5. Automatic hidden line removal occurs only between components detailed within a single command. The automatic process removes overlapping lines between shapes that are in front of each other. It does not remove intersections of shapes, that is, one shape completely or partly inside of the other. The coplanar surfaces of shapes which comprise an item are blended (for example, lines within the surfaces are hidden).

The hiding performed with the modifier REPLACE does not take into account any hiding done previously with HIDE PIPE. The detail graphics generated by GENERATE DETAIL REPLACE display only the hidden lines automatically removed because of the overlapping of components detailed together.

6. The clearance representations generated with the modifier CLEARANCE are either identical to, or a close approximation of, the shape(s) used for clearance checking. For more information on clearance representations, see CHECK CLEARANCE in the *Reports and Analyses Reference*.
7. Identify each nodal figure individually when a component involves nested nodal figures, such as nozzles within pieces of equipment. You may use WIN or PWIN to do this.
8. Detail graphics for bends do not include weld graphics breaks at the transitions between bent and straight segments unless the bend has the SPECIAL property.
9. You may detail the same model entity in multiple views, but in only one view at a time. You can restore the graphics of the erased model entities by deleting all the associated details in a detailed view. Delete details by using the SELECT MODE DRAW VIEW and DELETE ENTITY commands. Alternatively, you can use REECHO ENTITY and select the model entity in a view where it is visible.
10. When you specify single line piping, the procedure for elbows is not used; elbows are represented symbolically. However, the defined procedures for other fittings are used for single line piping.
11. You may draft details with simple arc and string entities. First, use the SELECT MODE DRAW VIEW command. Then relate the detail graphics and model entities (Nlines and Nfigures) with the command
RELATE DETAIL.
12. ESC-Q aborts or breaks out of the command when used before this message appears: n COMPONENTS BEING SOLVED. The message indicates that the detail graphics have been generated and are about to be hidden against one another. After the point of graphics generation, the hiding process must be completed even if you use ESC-Q. The detail graphics generated before you escape are not displayed. Use the REGENERATE GRAPHICS command to display them. Use HIDE PIPE for hidden line removal.

13. The hidden lines removed from detail graphics involve the blanking of intervals on the entities (arc and strings) that make up the graphics. To remove all hidden lines, even those resulting from the component being hidden against itself, use the REGENERATE GRAPHICS NOFONT command. Remember to use the REGENERATE GRAPHICS FONT command when finished.
14. Use the command PACK DBASE after long GENERATE DETAIL runs. This results in a significant saving of storage space with little extra processing time.
15. When using the VWIN getdata modifier to identify a whole batch of entities, you may exceed the number of entities getdata can identify at one time. Since you are only interested in Nfigures and Nlines with associated procedures and pipe or duct Nlines, there is no need to overburden getdata with other entities. To prevent them from being needlessly identified, make them invisible by echoing off the layer they are on, blanking them, erasing them, or echoing them off. The MARK ENTITY command will help you identify entities, such as internal Nlines, to be rendered invisible.

GENERATE DLINE (Double Line)

Description

This command generates a double-line representation of a duct model. Both ducts and duct fittings are graphically represented. The schematic diagram of the model is erased and replaced with detailed graphics. The detailed shapes are made up of ducts and duct fittings which belong to the active drawing and the selected view.

Valid Modes

Draw mode with a view selected. Activate this mode with the following command:

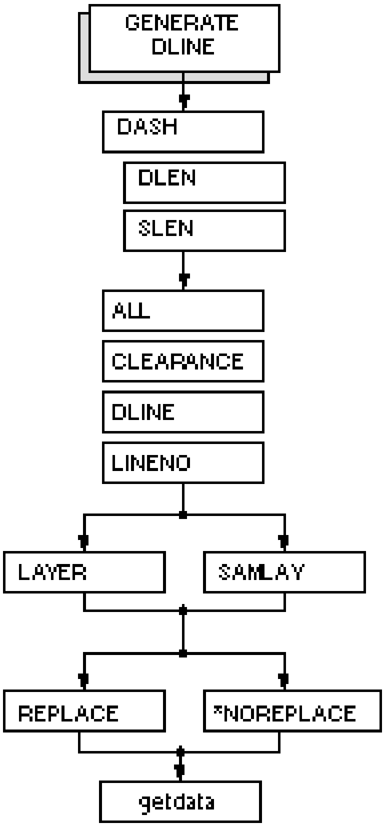
```
#n# SELECT MODE DRAW VIEW: view d1
```

Syntax

```
#n# GENERATE DLINE [modifiers]: <getdata>
```

Modifiers

The following diagram shows the flow of GENERATE DLINE modifiers. An asterisk (*) denotes a default modifier. For a full explanation of modifier diagrams, see explanation on “Modifier Diagrams” on page -xi in the Preface of this book.



Modifiers are listed alphabetically by level. When there are multiple levels of modifiers, level is indicated by the left number. A description of each modifier follows.

Please note: You can use the command SELECT PPARAMETERS to set or change the default values of some of the modifiers (where noted) for this command. Refer to the *Heating, Ventilating, and Air Conditioning Reference* for more information.

1 ALL: Generates a double-line representation for all entities. If more than one view exists in the drawing and no view has been selected, you are prompted to select a view.

1 CLEARANCE : Generates clearance shape representations for selected entities. If items already have appearance representations (from a previous use of

GENERATE DLINE), you must use the REPLACE modifier to replace them with clearance graphics.

1 DASH: Displays hidden lines as dashed lines (standard FONT DASHED). The default blanks out the hidden lines.

(Can be set by SELECT PPARAMETERS.)

2 DLEN x: Dash Length. Specifies the length of a dash in drawing units. The range is 0 to 32767. The default is 1.

(Can be set by SELECT PPARAMETERS.)

2 SLEN x: Space Length. Specifies the distance between dashes. The default is 1.

(Can be set by SELECT PPARAMETERS.)

1 DLINE: Ductline. Details every duct and fitting in the identified ductlines. Stand-alone components such as equipment, must be selected. Each selection on a pipeline component identifies all components in the ductline.

1 LAYER n: Inserts rendered graphics on layer n. You can use 0 to 254. The default is the active layer.

(Can be set by SELECT PPARAMETERS.)

1 LINENO text: Line number. Details a ductline with the specified line number. Only the first ductline found with the specified label is identified. No getdata is required or allowed.

1 NOREPLACE: (Default) Generates appearance or clearance representations for selected items not previously detailed.

1 REPLACE: Creates new double-line graphics and erases the old double-line graphics in the selected view. If this modifier is not specified, the double-line graphics are not updated.

1 SAMLAY: Same Layer. Creates detail entities on the same layer as the corresponding model entity. Without SAMLAY, detail entities go to the active construction layer or to the layer specified with the modifier LAYER.

Procedure 1: Generating Detail Graphics with Hidden Lines Removed

This procedure details nodal figures with hidden lines removed. You must select the nodal figures you want to detail. Only one selection is needed per nodal figure. You can use the getdata modifiers NLINE and NFIG to filter the model entity selection.

```
#n# GENERATE DLINE [modifiers]: MODEL ent d1... d
#n# ECHO ENTITY CNODE OFF
```

1. Enter GENERATE DLINE.
2. Enter any modifiers you wish to use.
3. Enter a colon. The `MODEL ent` prompt is displayed.
4. Select the nodal figures that you want to detail.
5. Press RETURN.
6. Enter ECHO ENTITY CNODE OFF.
7. Press RETURN.

Example 1 illustrates this procedure.

Procedure 2: Generating Detail Graphics with Hidden Lines Dashed

This procedure generates detail graphics for a duct run with hidden lines visible as dashed lines and nodal lines removed.

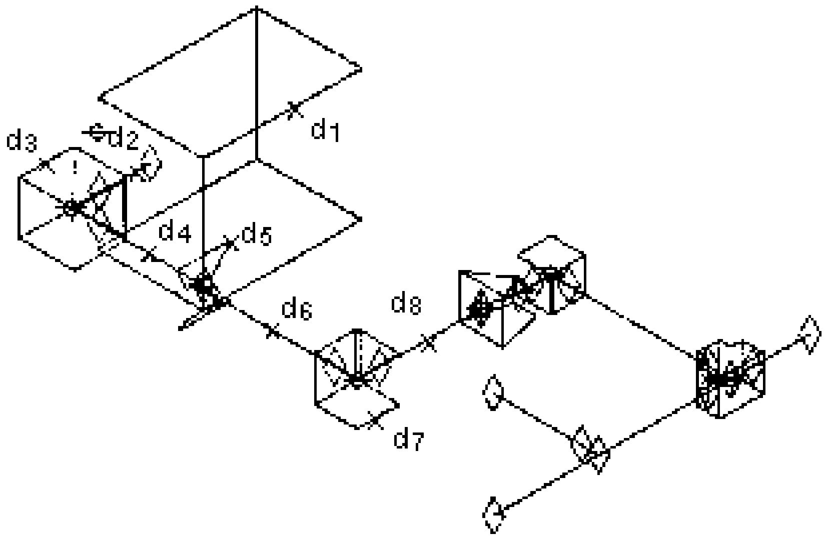
```
#n# GENERATE DLINE DASH [modifiers]: MODEL ent d1... d
#n# ECHO ENTITY CNODE NLINE OFF
```

1. Enter GENERATE DLINE.
2. Enter DASH and any other modifiers you want to use.
3. Enter a colon. The `MODEL ent` prompt is displayed.
4. Select the nodal figures that you want to detail.
5. Press RETURN.
6. Enter ECHO ENTITY CNODE NLINE OFF.
7. Press RETURN.

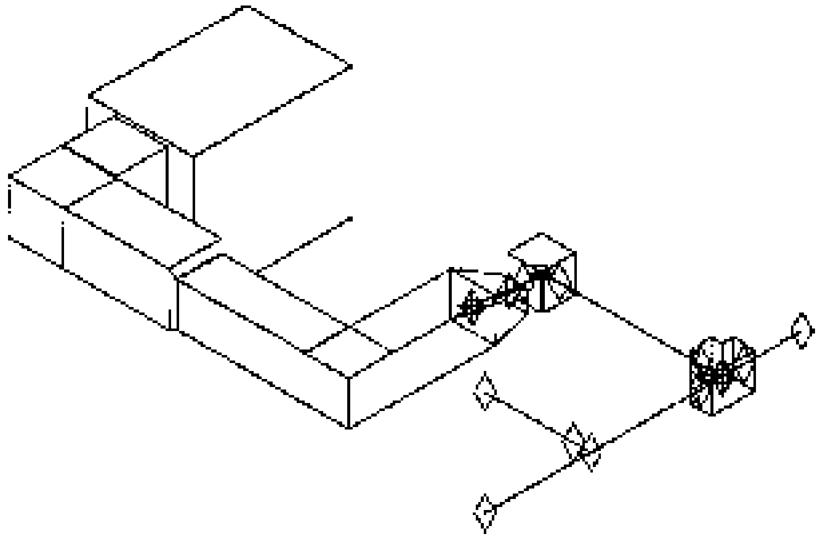
Example 2 illustrates this procedure.

Example1: The example on the following page shows the duct run created in the examples of the ROUTE DUCT and INSERT DFITTING commands (see the *Heating, Ventilating, and Air Conditioning Reference*). GENERATE DLINE is used on a portion of the duct run. The final detailed graphics are created with hidden lines removed.

```
#n# SELECT RELATIONSHIP ON
PARENT IDENT FLAG ON
#n# GENERATE DLINE: MODEL ent d1d2d3d4d5d6d7d8
8 COMPONENTS BEING SOLVED.
#n# ECHO ENTITY CNODE OFF
```



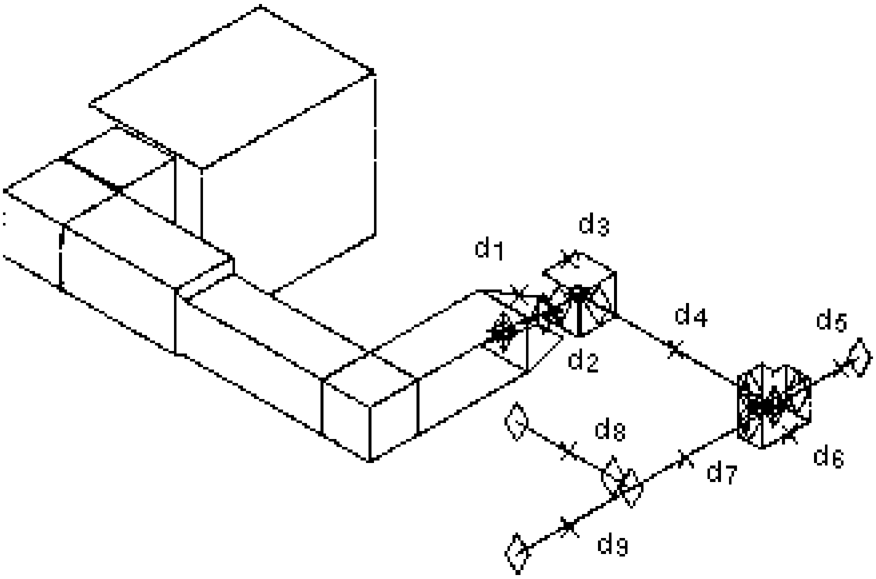
BEFORE



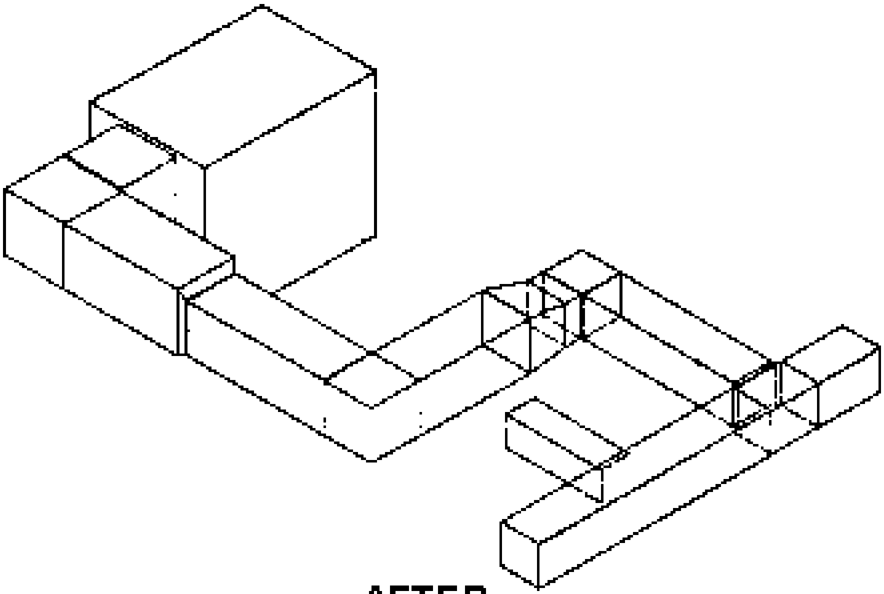
AFTER

Example 2: In this example, the remainder of the duct run from Example 1 is rendered with hidden lines dashed.

```
#n# SELECT RELATIONSHIP ON
PARENT IDENT FLAG ON
#n# GENERATE DLINE DASH SLENGTH.125 DLENGTH.25: MODEL ent
d1d2d3d4d5d6d7d8d9
9 COMPONENTS BEING SOLVED.
#n# ECHO ENTITY CNODE NLINE OFF
```



BEFORE



AFTER

Notes

Consider the following when using GENERATE DLINE:

1. GENERATE DLINE applies only to model nodal figures and nodal lines.
2. To render a nodal figure, there must be a procedure file associated with the figure. The default procedure file name is the nodal figure name.

You can also indicate which procedure file to use by inserting a PNAME property onto the nodal figure. The PNAME property stores the name of the procedure file. This is necessary only if you want a name that is different from the nodal figure name.

The parameters required by the procedure to render a fitting can be retrieved by looking up the stock number of the fitting in the parameter file. This happens only if the PLIST property is not present. If the PLIST property and a series of valves are present, those valves will be used to detail the nodal figure.

3. After performing any dynamic operation (DYNAMIC VIEW) on a view, issue the SET VIEW or RESET VIEW command. This prevents subsequent detailings from being lost from a temporary view when you file the part.
4. To identify individual ducts and duct fittings when you execute other commands, first use:

```
SELECT RELATIONSHIP OFF
SELECT MODE DRAW VIEW: d1
```

To identify entire detailed shapes when you are executing other commands, first use SELECT RELATIONSHIP ON in Model mode to relate detailed shapes to their nodal figures. Do not use this mode with MERGE RENDERING. When you are prompted `MODEL ent`, select anywhere on the detailed nodal figure.

5. Detailed shapes do not survive operations on the related nodal figure if the operation invalidates their appearance. For example, if you translate a nodal figure, its detailed graphics go with it. However, if you stretch, delete, break, or rotate a nodal figure, its detailed graphics are deleted.
6. Automatic hiding occurs only between nodal figures detailed within a single execution of the command. The command hides only the lines of overlapping shapes, that is, when one is in front of the other. It does not hide the lines of intersecting detailed shapes, that is, when one is partly inside the other.

Use the HIDE PIPE command to hide lines in nodal figures detailed in separate executions of GENERATE DLINE.

7. For each execution of the command, only new detailed shapes are hidden. Old detailed shapes from previous executions of GENERATE DLINE remain unchanged.

8. Detailed shapes only simulate a three-dimensional appearance. Because they are for appearance only, detailed graphics cannot be used in getdata with point-on-entity techniques.

9. You can detail the same nodal figure in multiple views, but in only one view at a time.

10. There are two ways to regenerate original nodal figures:

Use the DELETE VIEW command and then define a new view with the identical name and parameters.

Delete all the associated rendered entities in a rendered view. Use the commands SELECT MODE DRAW VIEW and DELETE ENTITY.

11. There are two ways to make the model entity visible in the given view:

Use the REECHO ENTITY command and select the model entity in a view where it is still visible.

Use the REGENERATE GRAPHICS NOFONT command.

An index file is accessed every time GENERATE DLINE is used with parameter file lookup. This index file is updated automatically when the index file date is less recent than the text file date. If you do not have write permission to write the updated index file into the library, an error message is displayed and detailing does not take place.

HIDE PIPE

Description

This command is primarily used to remove hidden lines within the piping details created by the GENERATE DETAIL command. Use HIDE PIPE to reflect the hiding of one item by another. For example, if a pipe crosses in front of a fitting, some lines within the detail graphics of the fitting must be hidden to give a realistic, three-dimensional appearance.

GENERATE DETAIL automatically does the hiding between items detailed in a single command. HIDE PIPE lets you do this plus additional hiding interactively for items detailed in separate executions of the GENERATE DETAIL command.

HIDE PIPE is also useful for touching up detail graphics directly. For example, it can be used to hide detail graphics and to produce cutaway sections.

Valid Modes

Draw mode with a view selected. To activate this mode use the following command:

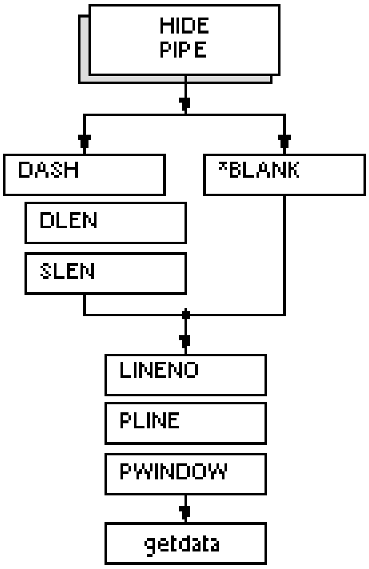
```
#n# SELECT MODE DRAW VIEW: view d
```

Syntax

```
#n# HIDE PIPE [modifiers]: <getdata>
```

Modifiers

The following diagram shows the flow of HIDE PIPE modifiers. An asterisk (*) denotes a default modifier. For a full explanation of modifier diagrams, see Notes to Readers.



Modifiers are listed alphabetically by level. When there are multiple levels of modifiers, level is indicated by the left number. A description of each modifier follows.

1 BLANK: (Default) Removes hidden lines by blanking them.

1 DASH: Displays hidden lines as dashed lines instead of blanking them. The default font (DASHED) is used.

2 DLEN n: Dash Length. Specifies dashing and the length of a dash as a positive real number in drawing units. The default is 0.16 inches regardless of the drawing unit. For example, this is listed as 4.06 in a part with millimeter units.

2 SLEN n: Space Length. Specifies dashing and the length of blank segments between dashes as a positive real number in drawing units. The default is 0.16 inches regardless of the drawing unit. For example, this is listed as 4.06 in a part with millimeter units.

1 LINENO text: Line Number. Identifies a pipeline by line number. The line number may include any adjacent fields in the pipeline label, for example, CW-100. Only the first pipeline found with the specified line number is identified. All pipes and fittings in the identified pipeline are hidden against each other.

1 PLINE: Pipeline. Hides all pipes and fittings on identified pipelines against each other. In response to the prompt `MODEL ent`, select a pipe on each pipeline.

1 PWINDOW: Polygon Window. Within a specified polygon window, hides all detail graphics related to the identified items. In response to the prompt `MODEL ent`, identify the items (Nlines, Nfigures). Enter a semicolon. Then in response to the prompt `MODEL loc` define the window. Note that items are not hidden against each other. This modifier is particularly useful for touching up detail graphics directly. In effect, HIDE PIPE acts as an erasing shield.

The maximum number of polygon points is 20; the minimum is 3.

Procedure 1

#n# SELECT RELATIONSHIP ON

```
#n# HIDE PIPE [modifiers]: MODEL ent d1...dn
```

1. Enter SELECT RELATIONSHIP ON to identify erased model entities with detail graphics.
2. Enter HIDE PIPE and any modifiers you want to use.
3. After typing a colon, respond to the prompt `MODEL ent` by selecting any one of the details related to each item (Nfigure or Nline) you want to hide.
4. Press RETURN.

Procedure 2

#n# SELECT RELATIONSHIP ON

```
#n# HIDE PIPE [modifiers]: MODEL ent d1...dn; MODEL ent WIN IN
CROSS d2 d3
```

1. Enter SELECT RELATIONSHIP ON to identify erased model entities with detail graphics.
2. Enter HIDE PIPE and any modifiers you want to use.
3. After typing a colon, respond to the prompt `MODEL ent` by identifying a first set of items. (Select anywhere on the detail graphics related to the items.) These are any new or additional items you want to hide. Then enter a semicolon and identify a second set of items with a window (WIN IN CROSS). Usually these are old or previously detailed items to be hidden against the first set.

The window is not required but is convenient, given the likelihood that set two involves all the items that overlap the visual extents of set one.

The items in the first set are hidden against each other and against those in the second set. Items in the second set are not hidden against each other during this procedure; it is presumed that such hiding has already been done.

This procedure saves significant time (as compared to procedure one) when a set of items (set two) has already been hidden against itself and functions as a background for hiding newly detailed items.

4. Press RETURN.

Procedure 3

#n# SELECT RELATIONSHIP ON

```
#n# HIDE PIPE PWINDOW [modifiers]: MODEL ent d1...dn; MODEL loc
d2...dn
```

1. Enter SELECT RELATIONSHIP ON to identify erased model entities with detail graphics.
2. Enter HIDE PIPE and PWINDOW and any other modifiers you want to use.
3. After typing a colon, respond to the prompt `MODEL ent` by selecting any one of the details related to each item (Nfigure or Nline) you want to hide.

4. Enter a semicolon after you have identified all the items you want to process. Respond to the prompt `MODEL loc` by specifying model locations of a polygon window covering the detail graphics you want to hide.
5. Press RETURN.

Please note: Use the NLIN and NFIG modifiers in getdata to filter the entity selection.

Example 1

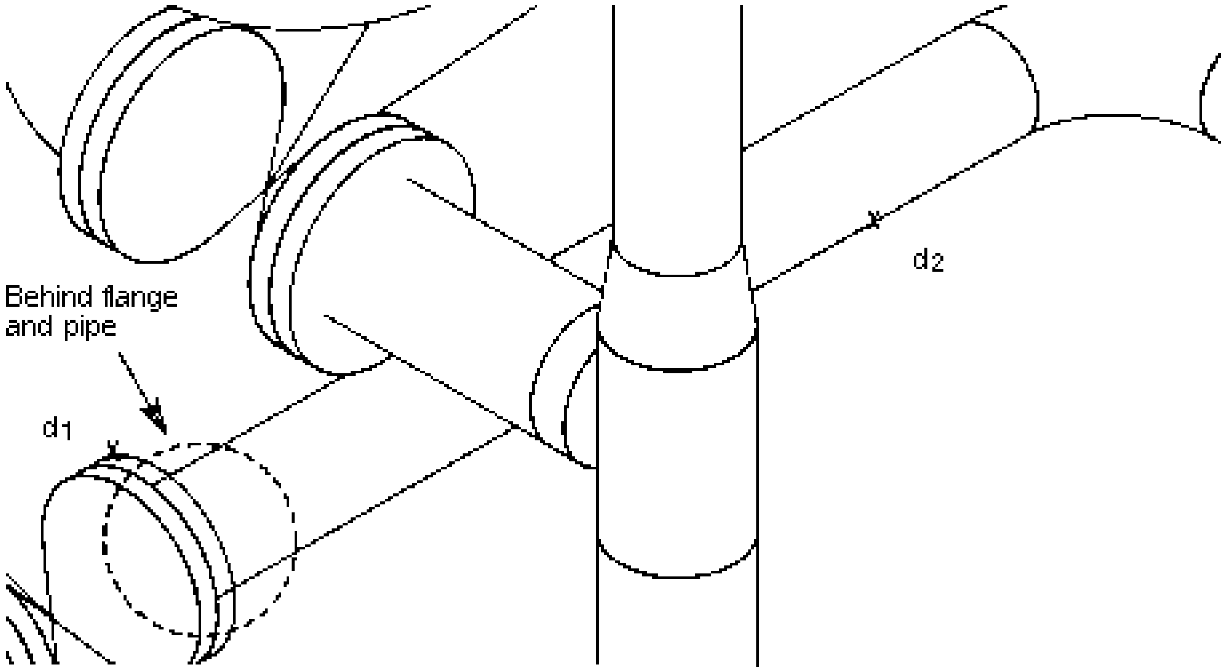
For HIDE PIPE to work, the model entities must be identified indirectly by selecting any one of the details related to them. To do this, first use the following command:

```
#n# SELECT RELATIONSHIP ON
PARENT IDENT FLAG ON
Then input this command to activate Draw mode with a view named
IS02 selected.
#n# SELECT MODE DRAW VIEW: view NAME ISO2
SELECTED MODE IS DRAW
```

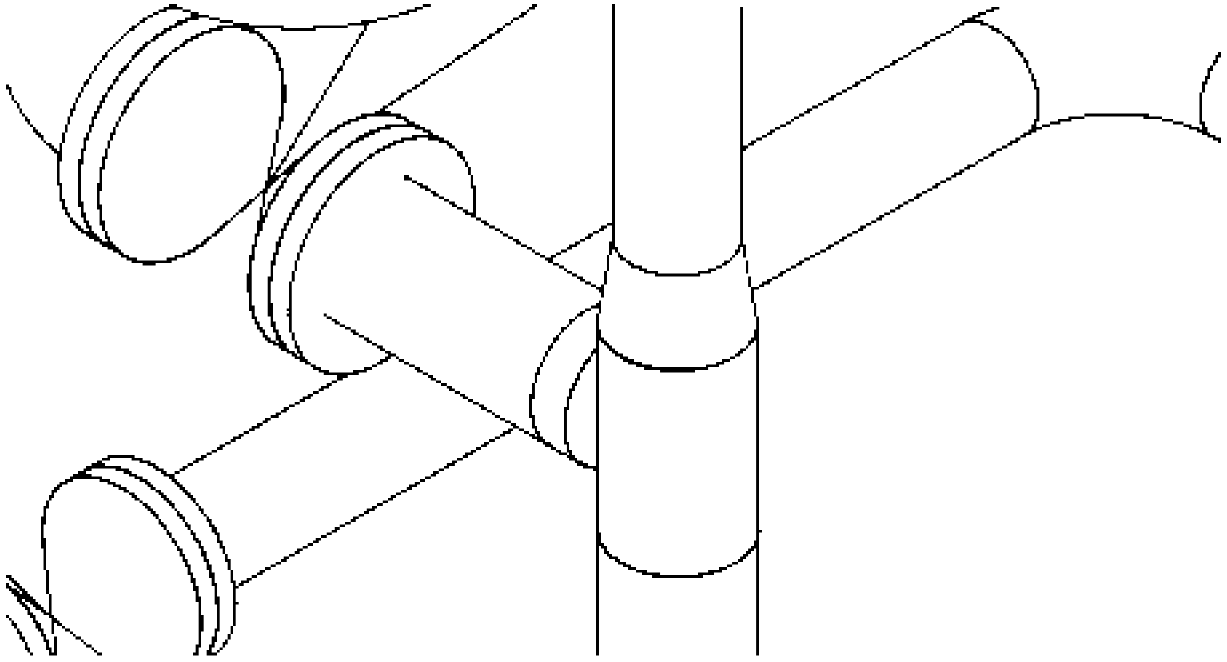
The model entities can now be reached through the detail graphics.

The following basic use of the command allows you to effect hiding between individual entities that were not detailed in the same execution of the GENERATE DETAIL command.

```
#n# HIDE PIPE:MODEL ent d1 d2
2 COMPONENTS BEING SOLVED
```



BEFORE

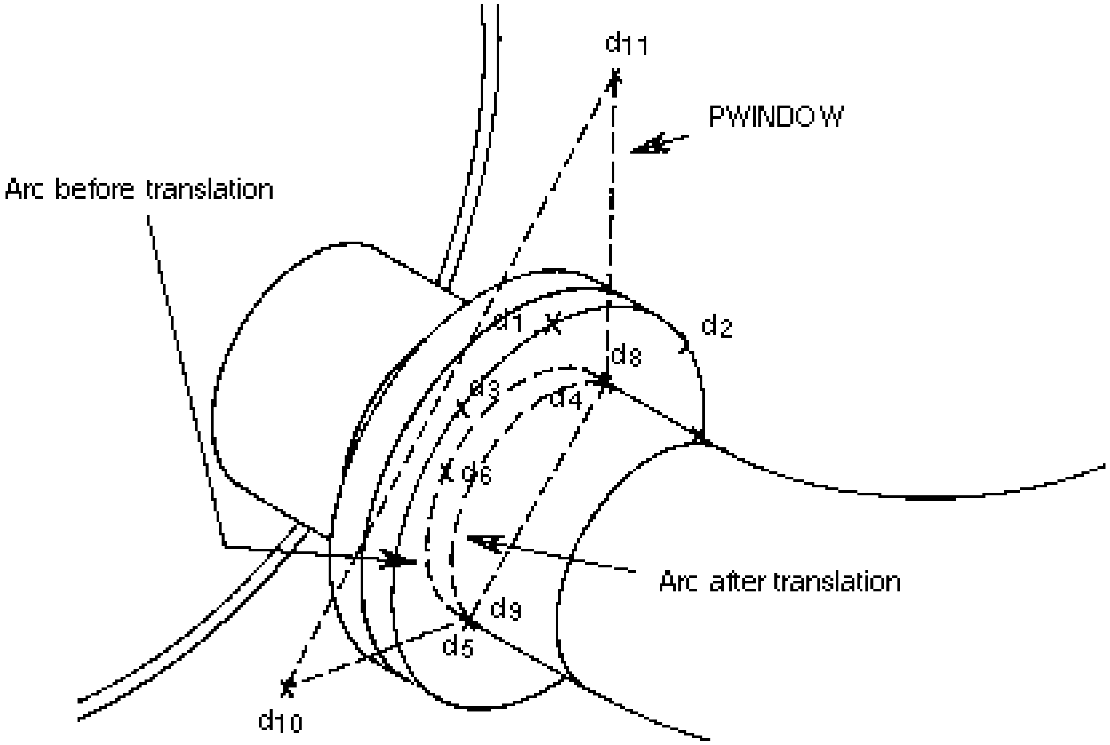


AFTER

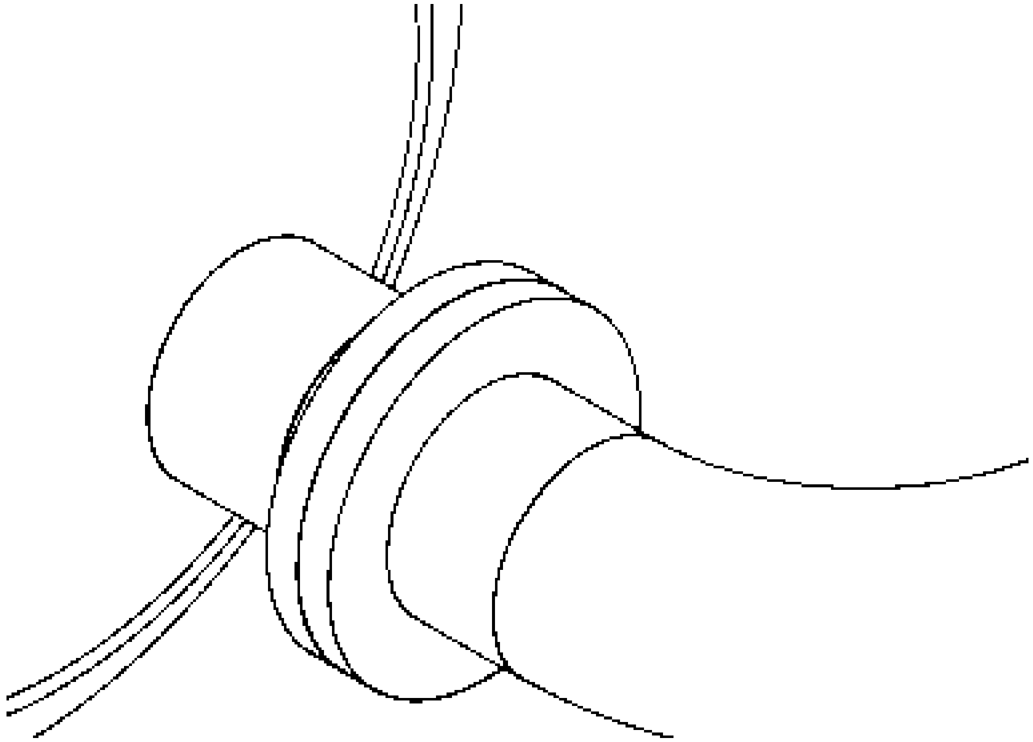
Example 2

This example uses the PWINDOW modifier to cut back the pipe detail which extends too far into the slip on flange. First, you detail the end of the pipe with an arc. You should relate the arc to the pipe Nline. This is shown in an example of the RELATE DETAIL command (see online command reference documentation). Then use the ends of the arc as points from which to cut back the pipe detail. (The dashed lines are shown here for demonstration purposes only.)

```
#n# SELECT MODE DRAW VIEW: view NAME BEFORE
SELECTED MODE IS DRAW
#n# DEFINE CPL TEMP: MODEL loc ORG ARC d1 NEAR d2d3
#n# INSERT ARC DIAMETER: MODEL loc CPL NAME TEMP VERT STR d4d5
#n# TRANSLATE ENTITY: DRAW ent d6; ORG ARC d6d1
#n# SELECT RELATIONSHIP ON
PARENT IDENT FLAG ON
#n# HIDE PIPE PWIN: MODEL ent d7;MODEL loc END ARC d8d9 LOC d10d11
```



BEFORE



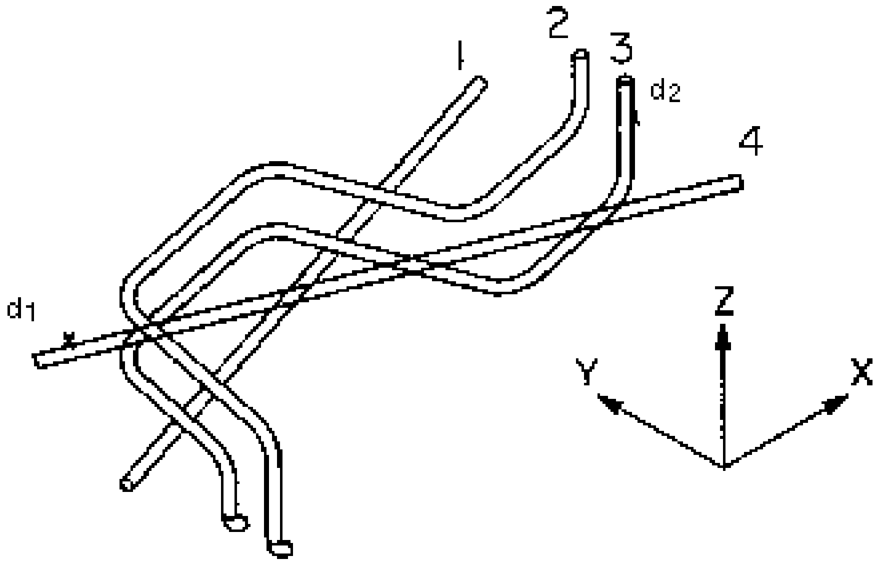
AFTER

Example 3

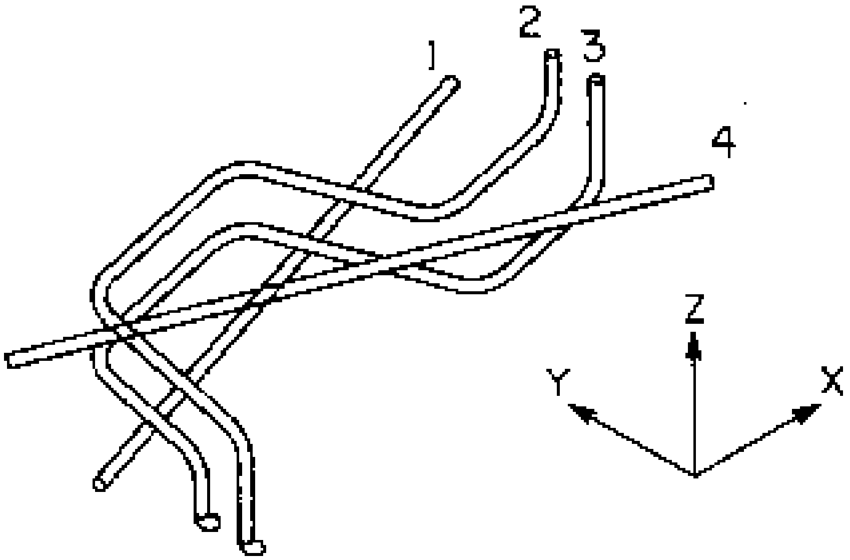
Pipeline 4 was added after detailing was done on the other pipelines. Notice that pipeline 3 crosses pipeline 4 in several places. To hide 3 against 4, use the PLINE modifier and only select each pipeline (Pline) once, rather than selecting the individual pipe entities at each crossing. PLINE is useful even with only one pipeline when it crosses itself.

```
#n# HIDE PIPE PLINE: MODEL ent d1d2
13 COMPONENTS BEING SOLVED
```

Notice the places where pipeline 3 crosses pipelines 1 and 2 that were hidden before. They remain intact after running the above command. There are still crossings of the new line 4 and lines 1 and 2. These can be hidden individually, but the next example shows a better way.



BEFORE

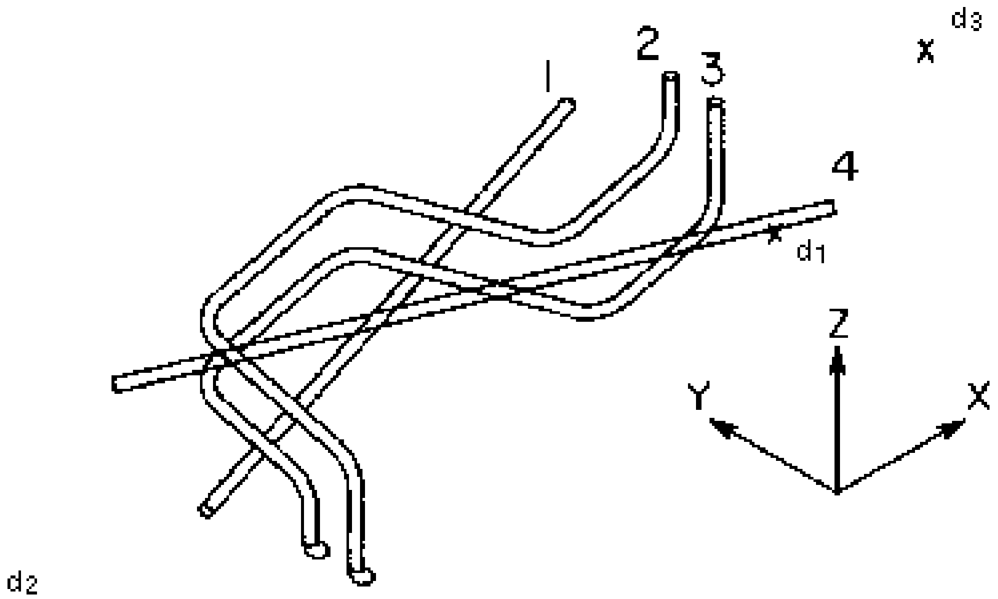


AFTER

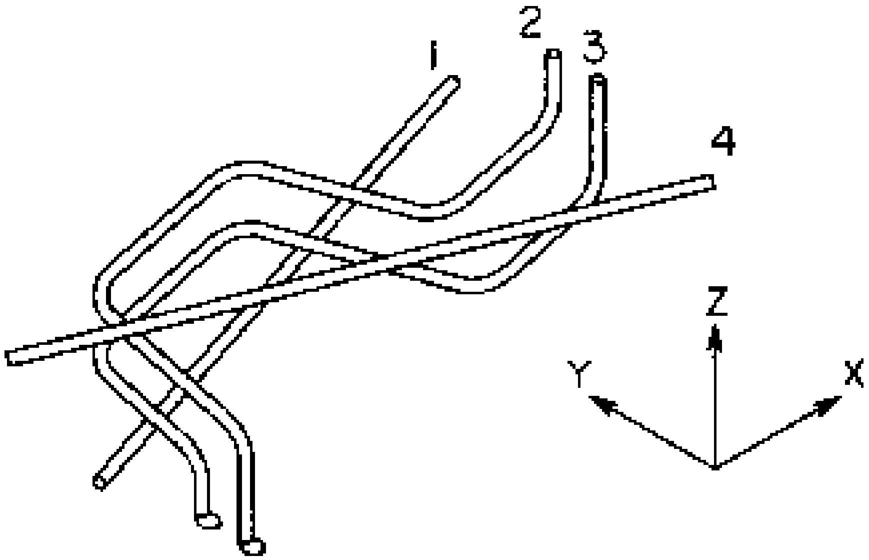
Example 4

Using the semicolon, you can divide entities into sets. The second set acts as a background against which the first set is hidden. The best way of hiding a line that is added after others are detailed is to treat it as set one. This saves time because the items in set two are not rehidden against each other.

```
#n# HIDE PIPE: MODEL ent d1; MODEL ent WIN DRAW loc IN CROSS d2d3
26 COMPONENTS BEING SOLVED
```



BEFORE



AFTER

Notes

The following general notes will help you further in using the HIDE PIPE command.

1. HIDE PIPE works only on entities that are created in Draw mode with a selected view and that are related to a model Nfigure or Nline. Entities created by the GENERATE DETAIL command automatically satisfy this restriction. The procedure file used in their generation is not used by HIDE PIPE.

The command also works on entities created interactively. You may draft details interactively by inserting arc or string entities in Draw mode with a view selected. You then relate them to an Nline or Nfigure with the RELATE DETAIL command (see the online command reference documentation).

If the command is used on an unscaled isometric, the model geometry stored in the XYZ property is automatically referenced.

2. Hiding of lines between components does not apply to so-called intersections between appearance shapes, that is, when one is partly inside the other. Hiding only takes into account the overlapping between shapes, that is, when one is in front of the other.
3. Although the commands HIDE PIPE and CHANGE APPEARANCE INTERVAL BLANK are similar, do not use both on the same detail graphics. The blank intervals the commands create can interfere with each other. For more information on CHANGE APPEARANCE, see the online command reference documentation.

MERGE RENDERING

Description

This command erases hidden lines of shapes that were rendered with different executions of the CREATE RENDERING command (see CREATE RENDERING). Whereas CREATE RENDERING automatically erases hidden lines of nodal figures rendered in a single execution, MERGE RENDERING erases additional lines that may become hidden by overlapping figures that were rendered in separate executions of CREATE RENDERING.

Valid Modes

Draw mode with a view selected. Activate this mode with the following command:

```
#n# SELECT MODE DRAW VIEW: view d1
```

In Model mode, specify a view with hidden line removal using the following command:

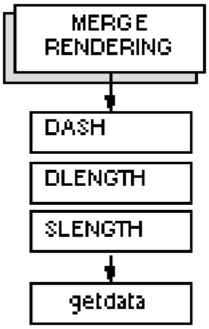
```
#n# SELECT MODE MODEL
```

Syntax

```
#n# MERGE RENDERING [modifiers]: <getdata>
```

Modifiers

The following diagram shows the flow of MERGE RENDERING modifiers. For a full explanation of modifier diagrams, see Notes to Readers.



Modifiers are listed alphabetically by level. When there are multiple levels of modifiers, level is indicated by the left number. A description of each modifier follows.

DASH: Displays hidden lines as dashed lines when these lines are on a face that is only partially invisible. When entire faces of rendered graphics are invisible, no dashes are generated. The standard line font DASHED is used.

DLENGTH x: (Dash Length) Specifies the length of a dash as a positive real number in drawing units. The range is 0.0 to 32,767.0. The default is 1.0.

SLENGTH x: (Space Length) Specifies the length of blank segments between dashes as a positive real number in drawing units. The default is 1.0.

Procedure

#n# SELECT RELATIONSHIP ON

```
PARENT IDENT FLAG ON
```

```
#n# MERGE RENDERING [modifiers]: MODEL ent d1... dn
```

1. Enter SELECT RELATIONSHIP ON to relate nodal figures to their rendered shapes.
2. Press RETURN. The system prompts `PARENT IDENT FLAG ON.`
3. Enter MERGE RENDERING.
4. Enter any modifiers you wish to use.
5. Enter a colon.
6. The `view` prompt appears if you previously selected Model mode with the command:

#n# SELECT MODE MODEL

Select the view(s) you want rendered.

7. After the `MODEL ent` prompt, select Nfigures that you rendered with separate executions of CREATE RENDERING. Even though each individual Nfigure looks three-dimensional with hidden lines removed, figures that are partially covered by other figures lack all necessary hidden line removal. Select these unfinished Nfigures to erase all hidden lines.
8. Press RETURN.

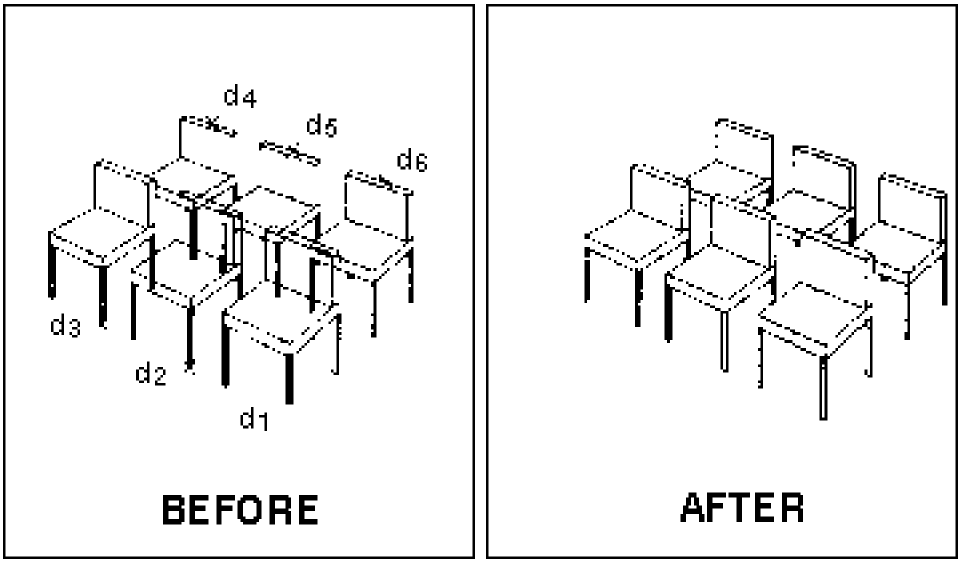
All hidden lines are erased from the selected rendered shapes that overlap other shapes so that the entire view appears solid.

Example

This example shows the erasing of hidden lines when Nfigures are rendered with separate executions of CREATE RENDERING, leaving some hidden lines still visible.

First, a chair Nfigure was inserted three times and rendered. Second, three additional copies of the chair were inserted and rendered (perhaps these chairs were originally inserted at a different location and then translated). The front row of chairs now partially overlaps the back row. Third, MERGE RENDERING is used to erase the hidden portions of the rear chairs.

```
#n# SELECT RELATIONSHIP ON
PARENT IDENT FLAG ON
#n# MERGE RENDERING: MODEL ent d1d2d3d4d5d6
```



Notes

Consider the following when using MERGE RENDERING.

1. MERGE RENDERING works on entities created by the CREATE RENDERING command and on other nodal figures that use strings representing planar faces of objects such as projections (created by the command PROJECT OUTLINE), and projected polygons (created by the command PROJECT POLYGON).

MERGE RENDERING also works on entities you add interactively. First, use the command SELECT MODE DRAW VIEW. Second, use INSERT STRING and INSERT ARC. Third, use the command RELATE DETAIL to relate the rendered entities to the original nodal figures.

2. You may use CREATE RENDERING to perform hidden line removal for nodal figures that use strings representing planar faces of objects (such as walls, projections, and steel beams). MERGE RENDERING is all that is needed for hidden line removal on steel objects.
3. Before using the command, you must execute SELECT RELATIONSHIP ON to relate original nodal figures to their renderings.
4. After MERGE RENDERING, use SELECT RELATIONSHIP ON to create rendering or merge rendering on other entities.

5. Use SELECT RELATIONSHIP ON only for procedure-file-based entities (not for projected outlines, walls, or displayed beams).
6. Note the special use of the semicolon in this command to increase the speed at which the command works. Using a semicolon in getdata between groups of selected objects performs hidden line removal between the entire groups of entities. However, this method does not perform hidden line removal between the objects within each group.

UNHIDE PIPE

Description

This command redisplays hidden detail lines that were hidden with the HIDE PIPE or GENERATE DETAIL commands.

Valid Modes

Draw mode with a view selected. To activate this, use the following command:

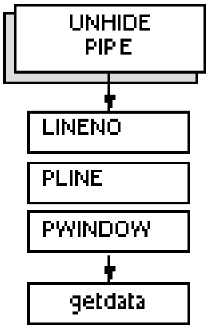
```
#n# SELECT MODE DRAW VIEW: view d
```

Syntax

```
#n# UNHIDE PIPE [modifiers]: <getdata>
```

Modifiers

The following diagram shows the flow of UNHIDE PIPE modifiers. For a full explanation of modifier diagrams, see Notes to Readers.



Modifiers are listed alphabetically by level. When there are multiple levels of modifiers, level is indicated by the left number. A description of each modifier follows.

LINENO delimited text: Line Number. Specifies the line number of a pipeline to process. All hidden lines in the pipeline's detail graphics are redisplayed. The line number may include any adjacent fields in the pipeline's label, for example, CW-100. Only the first pipeline found with the specified line number is identified.

PLINE: Pipeline. Redisplays all hidden lines in the detail graphics related to the identified pipelines. Respond to the prompt MODEL ent by selecting a detailed pipe on each pipeline to be processed.

PWINDOW: Polygon Window. Redisplays hidden lines in detail graphics within a specified polygon window. This redisplay applies only to the details related to identified model entities. Respond to the prompt MODEL `ent` by identifying the model entities. Then respond to the prompt MODEL loc by specifying the model locations for the polygon window.

The maximum number of polygon points is 20; the minimum is 3.

Procedure 1

#n# SELECT RELATIONSHIP ON

```
#n# UNHIDE PIPE: MODEL ent d1...dn
```

1. Enter SELECT RELATIONSHIP ON to identify erased model entities with detail graphics.
2. Enter UNHIDE PIPE.
3. After typing a colon, respond to the prompt `MODEL ent` by selecting any one of the details related to each item (Nfigure or Nline) you want to hide.
4. Press RETURN.

Procedure 2

#n# SELECT RELATIONSHIP ON

```
#n# UNHIDE PIPE PWIN: MODEL ent d1...dn; MODEL loc d2...dn
```

1. Enter SELECT RELATIONSHIP ON to identify erased model entities with detail graphics.
2. Enter UNHIDE PIPE and PWIN.
3. After typing a colon, respond to the prompt `MODEL ent` by selecting any one of the details related to each item (Nfigure or Nline) you want to hide.

4. Enter a semicolon after you have identified all the items you want to process. Respond to the prompt `MODEL loc` by specifying model locations of a polygon window covering the detail graphics with hidden lines you want to redisplay.
5. Press RETURN.

Please note: Use the modifiers NLIN and NFIG in getdata to filter the entity selection.

Example 1

For UNHIDE PIPE to work, you must identify the model entities indirectly by selecting any one of the details related to them. To do this, first issue the following command:

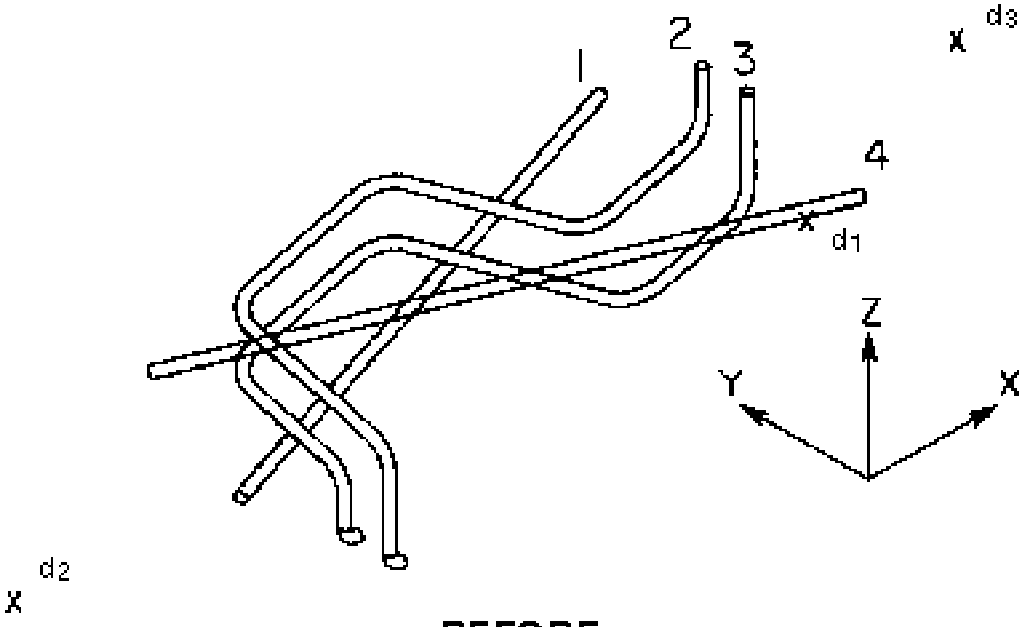
```
#n# SELECT RELATIONSHIP ON
PARENT IDENT FLAG ON
```

Then input this command to activate Draw mode with a view named ISO2:

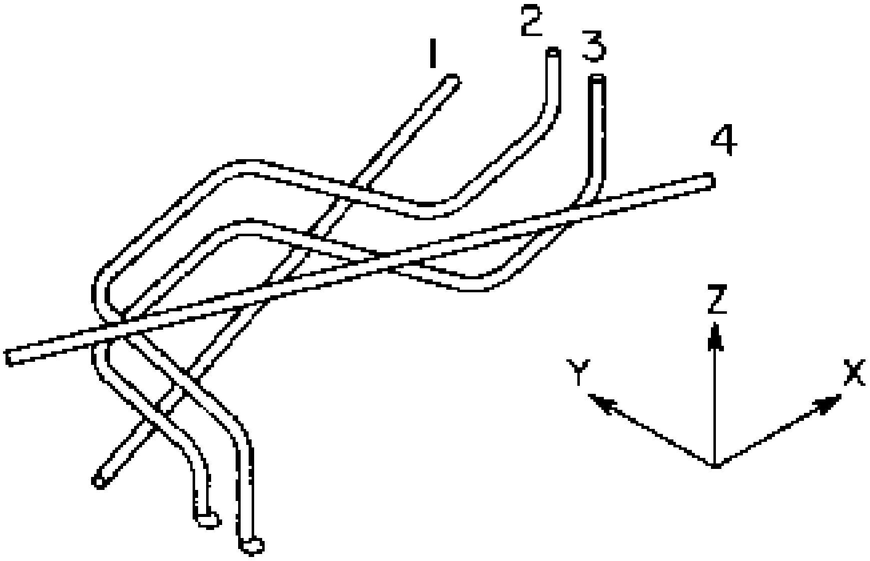
```
#n# SELECT MODE DRAW VIEW: view NAME ISO2
SELECTED MODE IS DRAW
```

The model entities can now be reached through the detail graphics.

```
#n# UNHIDE PIPE: MODEL ent d1d2d3d4d5d6
```



BEFORE

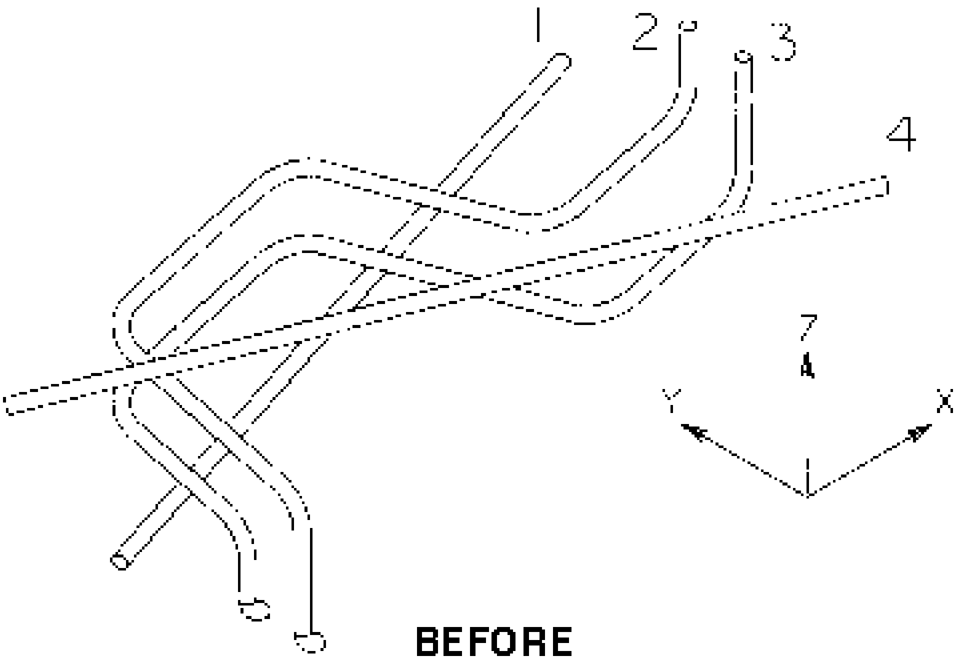


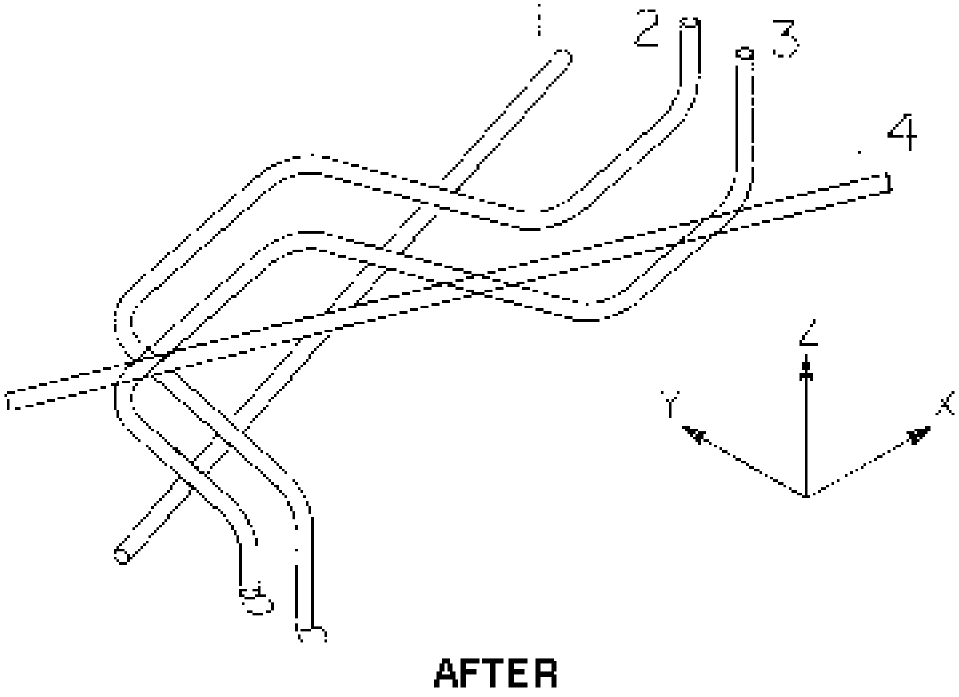
AFTER

Example 2

This example shows how to unhide all hidden portions of a pipeline. LINENO is used instead of selecting the line.

```
#n# UNHIDE PIPE LINENO 3
```

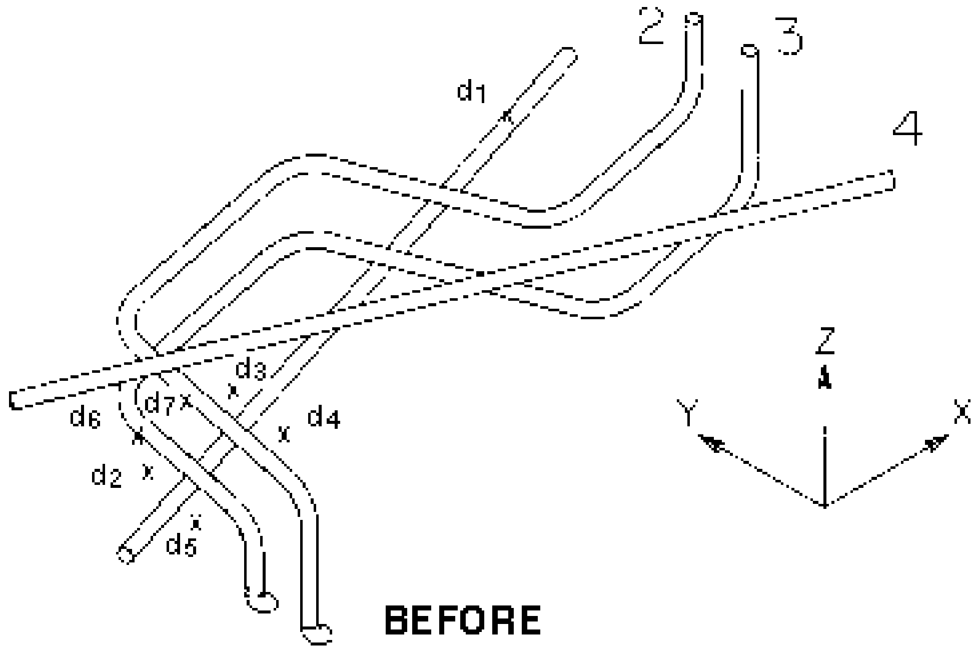


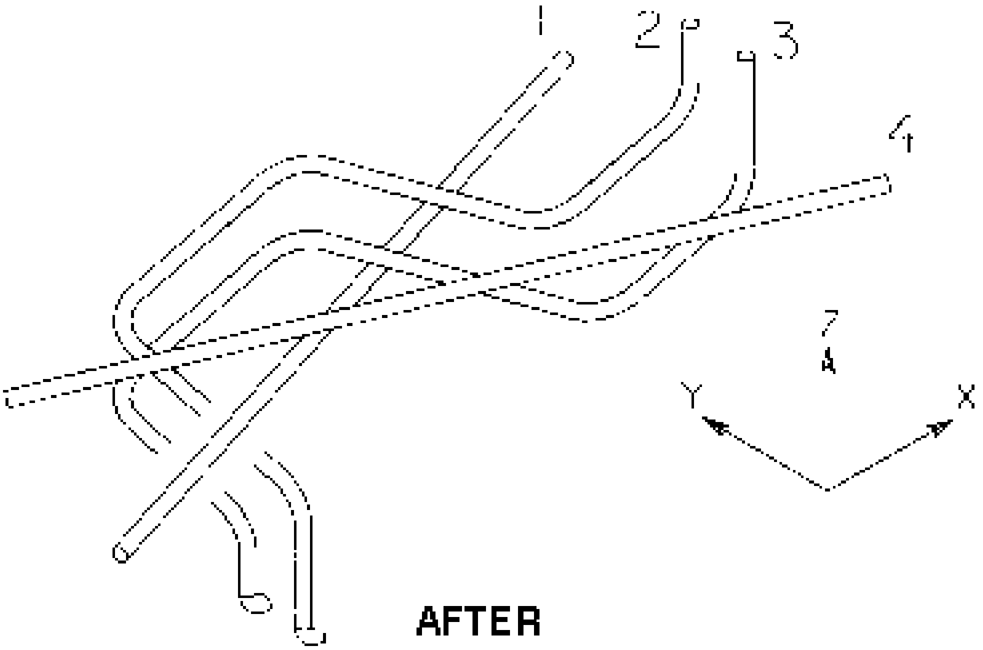


Example 3

This example uses the PWINDOW modifier. An item may be hidden in more than one place. If you want to remove only some of its hidden lines, use PWINDOW as shown below. Notice only the hidden lines of pipeline 1 at the intersections with pipelines 2 and 3 are unhidden; other hidden lines remain hidden. (Pipeline 4 is one pipe entity.) With HIDE PIPE, a cutaway is created to uncover pipeline 1 where it runs under pipelines 2 and 3.

```
#n# UNHIDE PIPE PWINDOW: MODEL ent d1; MODEL loc d2d3d4d5
#n# HIDE PIPE PWINDOW: MODEL ent d6d7; MODEL loc d2d3d4d5
```


2
3
d1
4
Z
d3
d7
d6
d4
Y
X
d2
d5
BEFORE


1
2
3
4
Z
Y
X
AFTER

Notes

Consider the following when using the UNHIDE PIPE command.

1. Some hidden lines cannot be redisplayed, for example, those hidden to create the three-dimensional appearance of an item hidden against itself.
2. If all of the individual entities forming an item's detail graphics are totally hidden in a view, you cannot identify the model entity in the view. This means that you cannot hide, unhide, replace, or even delete the entity's detail graphics. If some of the details of a model entity are visible and some are hidden, you can still identify the model entity. However, it remains erased in the given view even after you attempt to delete all its related details. The reason for this is that any details totally hidden cannot be deleted.

There are three ways to make the model entity visible in the given view:

- **a.** Use the REECHO ENTITY command (see the online command reference documentation*)* and select the model entity in a view where it is still visible.
- **b.** Use the REGENERATE GRAPHICS NOFONT command (see the online command reference documentation) and then delete the detail graphics.
- **c.** Use the DELETE FITTING DETAIL command (see the *Piping Reference*) to delete all the detail entities related to the model entity.

HIDE OBJECT

Description

This command performs hidden-line removal (HLR) on all identified solids, surfaces, curves and AEC objects in one or more views that you specify. It produces a complete three-dimensional scene or image. The HIDE OBJECT command if applied to previously HLRed entities, preserves the dimensions of Limbs or Surrogates. Similarly, if the dimensions are applied to the viewed part entities, HIDE OBJECT preserves the dimensions for those curves which are still visible after the execution of the command.

If dimensions are applied to the Surrogates in HLR Image, the UNHIDE OBJECT command preserves these dimensions and they are applied to the corresponding viewed part curves. Dimensions applied to Limbs are not transferred.

Valid Modes

Model mode only.

Syntax

Syntax for removing hidden lines of object(s):

```
HIDE OBJECT [modifiers]: view d1...dn; Model Ent d1...dn
<CR>
```

Syntax for removing hidden lines using Facetedengine along with the Quality modifier:

```
HIDE OBJECT Facetedengine Qualityfactor <getdata> [modifiers]:
view d1...dn<CR>
```

Modifiers

Modifiers are listed alphabetically by level. A description of each modifier follows.

ALL: Select all objects on all echoed layers of selected views. When multiple views are selected for HLR, objects in all the echoed layers in all the selected views are processed for HLR. The HLR results are displayed in each selected view as per the echoed layers of that view.

Please note: Up to Revision 8.1.1, ALL modifier selects all objects on echoed drawing layers.

The ALL modifier can be restricted to particular entity types by the following submodifiers:

- AEC: Selects all AEC objects on all echoed layers of selected views.
- CURVE: Selects all curves on all echoed layers of selected views.
- SOLID: Selects all solids on all echoed layers of selected views.
- SURFACE: Selects all surfaces on all echoed layers of selected views
- MARK: Includes or excludes marked entities from the selected entities.

  The MARK modifier has the following submodifiers:

  - INCLUDE: Specifies that from the list of selected entities, only the marked entities are processed for HLR.
  - EXCLUDE: Specifies that from the list of selected entities, only the entities that are not marked, are processed for HLR.
- CLIP: Selects the entities within and crossing the selected view boundaries along the X-, Y- and Z-axes of the view. You can also use the CLIP modifier with any of the four submodifiers AEC, CURVE, SOLID and SURFACE.

  The CLIP modifier has the following submodifiers:

- IN: Selects only the entities within the selected view boundaries.
- CROSS: Selects only the entities crossing the selected view boundaries.

Please note: To select the entities within and crossing the view boundaries, you can either use both the IN and CROSS modifiers or use neither of them.

BLANK: All hidden lines are blanked. This is the default. It cannot be used with the DASH modifier.

DASH: Displays hidden lines as dashed lines. The default dash font is DASH. You can use DLEN and SLEN to modify the dash font. It cannot be used with the BLANK modifier.

DLEN: Specifies the dash length in absolute drawing units. The default dash length is 0.25 drawing units.

FACETEDENGINE: Specifies the Faceted engine that approximates the object for HLR by tessellating it according to the Quality factor specified. The speed or quality of HLR operation will depend on the value specified for QUALITYFACTOR.

QUALITYFACTOR: Specifies a quality factor ranging from a low of 0 to a high of 100 and is inversely proportional to the speed of HLR image output. This modifier is available only while using the Facetedengine modifier.

Please note: The modifiers in the following list cannot be used along with FACETEDENGINE:

SLEN: Specifies the inter-dash spacing in absolute drawing units. The default dash spacing is 0.25 drawing units.

INTERFERENCE: Enables processing designed to minimize errors that may arise from the presence of interfering objects in the image. The use of INTERFERENCE is not generally recommended as it may degrade performance.

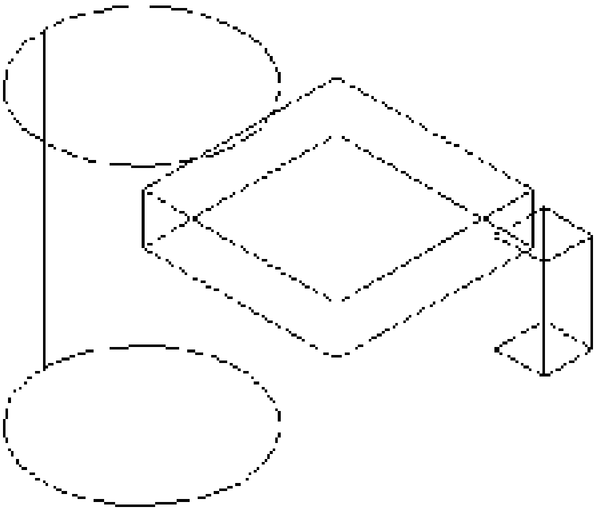
INTERSECT: Displays the intersection curves between objects. It automatically enables INTERFERENCE. This modifier involves a Boolean operation between the interfering objects and may take significant time for very complex and large assemblies and parts.

ILAYER: Specifies the layer in that the HLR intersection curves are displayed. If the layer is not specified, the active layer value is used.

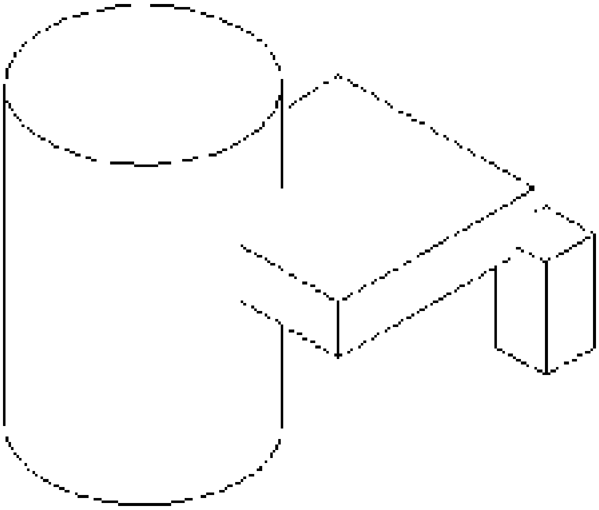
ITOL: Specifies tolerance to be used for the intersection process. If the tolerance is not specified, the default tolerance value is used.

Example

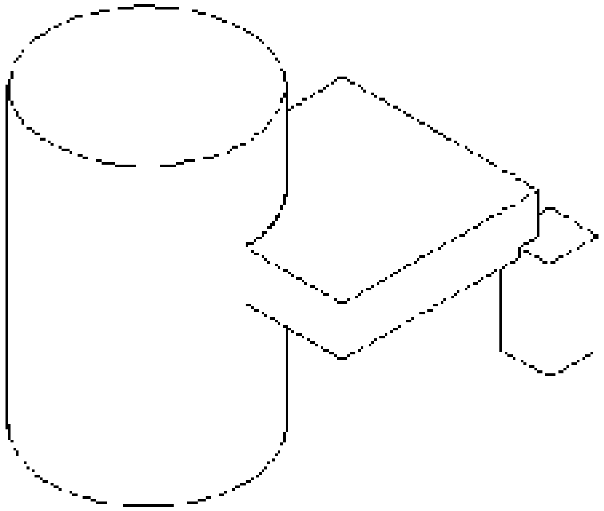
The first figure shows the input objects that are interfering with each other before using the HIDE OBJECT command.



The figure below shows the results after the using the HIDE OBJECT command.



The results of the HIDE OBJECT command, when used with the INTERSECT modifier, are as shown in the figure below.



LIMBTOL: Specifies the tolerance factor to be used for limb generation. The default value is derived from TOLERANCE. If the quality of the limbs is unacceptable, a smaller choice of LIMBTOL may improve the result. If limbs with an excessive number of Nurb control points are being generated, a larger LIMBTOL may help in reducing the database size.

MODFILE: Incorporates the components of the modifier file <filename> into the command. These components are executed in sequence after the modifier that precedes MODFILE, if any. For more information, see the *Explicit Modeling User Guide and Menu Reference*.

NONASSOCIATIVE: Creates the HLR image that is nonassociative to the model. Nonassociativity state is a property of the HLR image, indicating whether the surrogates that display the HLR picture are nonassociative or not.

REFINE: Provides better quality results of an HLR image.

SEGMENTS: Defines a global subdivision of curves for interference detection. The default value when INTERFERENCE is selected is SEGMENTS = 10. The use of SEGMENTS is not generally recommended as it may hamper the performance.

SHOWTOPOEDGES: Disables the suppression of all topological edges for the selected objects. A topological edge of a solid is the one that exists between two surfaces that are analytically identical.

TANGENCY: Enables the suppression of all tangent edges for the selected objects. A tangent edge of a solid is one for which the tangent planes of the faces sharing the edge match at every point along the edge.

TOLERANCE: Specifies a tolerance factor for HLR computations, including curve approximations and curve-curve intersections. The value is defined in terms of model units, unless DRAWUNITS is used.

If you do not select this modifier, a default TOLERANCE value is generated from view information.

DRAWUNITS: Specifies that the TOLERANCE value is in drawing units instead of model units.

VPVISIBLE: Overrides the default that causes viewed parts or assemblies to be blanked as part of the HIDE OBJECT process. If you specify this modifier, you must manually blank the viewed parts to obtain the correct HLR image. In the CAMU environment, the box image in a component node indicates that the component has an associated HLR image.

Notes

HIDE OBJECT and UPDATE HLRIMAGE preserve manually applied fonts to entities that are hidden.

UNHIDE OBJECT will remove all the fonts, in CAMU as well as in a piece part environment.

Restrictions

Certain restrictions apply when HIDE OBJECT is used on entities that have user-defined appearance information added before HIDE OBJECT is executed.

Appearance information includes:

- Font
- Blanked or fonted intervals
- Entity color
- Entity erasure

and it can be created using any of the following commands:

- INSERT <line/curve type> FONT
- INSERT <line/curve type> ECOLR
- CHANGE APPEARANCE FONT [ALLVIEWS]
- CHANGE APPEARANCE INTERVAL [ALLVIEWS]
- CHANGE ECOLR
- ERASE ENTITY

The result of executing a HIDE OBJECT command on entities with user-defined appearance information depends on whether or not the appearance information is specific to a view or views, or applies to all views. The FONT or INTERVAL information can be either view-specific or applied to all views. Entity COLOR always applies to all views, whereas entity erasure is always view-specific.

The intersection curves displayed by the INTERSECT modifier are nonassociative.

Restrictions and Notes

There are certain restrictions and other information you should be aware of.

User-Created Fonts are Removed

The system removes any previous user-created image data associated to an object in the view being used for HLR. To enhance the accuracy of the image appearance, Remove/Restore Hidden Lines eliminates any pre existing image data and creates new view specific font data on many of the curves and limbs associated with an object. If you use REECHO ENTITY after ERASE ENTITY on any fonted entity, then the font for the entity will not be preserved by HLR commands. The types of data that are removed by HLR are:

- View specific blanking or fonting of entities
- Incremental blanking or fonting of entities

Preservation of User-Created Fonts by HLR

HIDE OBJECT and UPDATE HLRIMAGE preserve manually applied fonts in CAMU as well as in viewed part environment.

Rotated Images are not Updated

If you rotate the image (using Rotate View, Dynamic View) on a view after using Remove Hidden Lines, you may rotate hidden lines to positions where they would no longer be hidden. The system does not automatically update the image with respect to the new viewing angle. If you rotate a view with hidden lines, you can restore the image with one of the following methods:

- Execute Restore Hidden Lines to restore the original objects.
- Execute Remove Hidden Lines again.
- Reset the view to its original orientation.

Surface Meshes are not Modified

Remove Hidden Lines does not modify surface meshes. You must change a mesh to 1 x 1 or blank any mesh that interferes with visualization of the hidden line image.

Warning

Before removing hidden lines, use Detect Interference to ensure that none of the objects penetrate each other. If the objects share common volume, the image created by Remove Hidden Lines could contain significant errors.

Using Remove Hidden Lines with Viewed Parts and Assemblies

Using Remove Hidden Lines on viewed parts (either from CAMU or Explicit View Part) creates surrogate copies of all visible edge curves. The command also blanks all viewed parts. At the end of the process, you have an HLR image in the views you designated.

In addition, these surrogate curves are grouped together, each group representing a single viewed object. This grouping prevents you from deleting an HLR curve by mistake. You should modify the appearance of the curve if it is not satisfactory.

To return the part to the original state, use the Restore Hidden Lines option and select the For all Objects button (digitize all HLR views). This will delete all the HLR groups in the active part and restore the viewed parts.

Using Remove Hidden Lines in Multiple Views

Performance is much better if you perform Remove Hidden Lines on all views at one time, particularly when working with viewed parts or assemblies. Doing so will reduce the number of entities created. Each execution of Remove Hidden Lines creates a new set of local curves; pre-existing HLR pictures will not be updated to reflect the creation of these curves.

Using Remove Hidden Line on Objects Created in the Parametric Environment

You can use Remove Hidden Lines on objects created in the Parametric environment and viewed in the Explicit environment. However, if you reenter the Parametric environment, change the object, and return to the Explicit environment, the hidden line graphics will be incorrect. You must re-execute Remove Hidden Lines.

Using Nonassociative HLR Image

Simultaneous display of exploded and unexploded views on one drawing sheet is achievable. When the view parts are unblanked, the nonassociative surrogates may be out of sync with the viewed (unhidden) wireframe.

Partial unhides are not permitted in nonassociative HLR Images. HLR picking and MARK CHANGE do not work for nonassociative HLR images.

Features that are supported through curve associativity, such as preservation of dimensions during updates, are not supported. The nonassociative HLR image is not deleted when deleting the corresponding component from the assembly tree structure. Picking the tree node does not highlight the corresponding nonassociative HLR image. nonassociative HLR images increase the database size.

Using Associative HLR Image

In order to assure that associativity is properly mapped between the parametric and explicit environments, the following line should be in the user's `.caddsrc` file:

```
setenv CADDSASSOC_UID yes
```

If a particular surface has more than one limb then all the dimensions become non associative and they are marked.

Please note: HIDE OBJECT can only maintain associativity to a viewed part when specific internal data exists in the viewed part. All CADDS design parts, except older CADDS parts, automatically contain this information. Such older parts have to be activated and filed as design parts in order to ensure subsequent associativity with respect to HLR.

Associating Viewed Components to Centerline, Label, or Hatch

- It will not be possible to associate an existing nonassociative centreline, label, or hatch with a viewed entity using CHANGE DIMENSION ASSOCITIVE command.
- If a part corresponding to a component is renamed or deleted, then the centreline, label, or hatch associated with the viewed component will be lost.

UPDATE HLRIMAGE

Description

This command re-executes hidden line removal (HLR) in one or more specified views. When no modifiers are selected with UPDATE HLRIMAGE the command uses the same modifiers and the same list of objects that were used in the last hide.

Please note: You must run the UPDATE HLRIMAGE command on an HLRed image in the same mode in which the image was created, that is, either in the standard mode or exploded mode. Otherwise, the command is aborted and the following warning message is displayed.

```
Update HLRimage command is not valid in this mode.
Please change the current adrawing orientation using
TOGGLE EXPLODED command and re-issue the UPDATE HLRimage
command.
```

Valid Modes

Model mode only.

Syntax

Syntax for updating HLR object(s):

```
UPDATE HLRIMAGE [modifiers]: view d1d2... dn <CR>
```

Syntax for updating HLR object(s) using the MERGE modifier:

```
UPDATE HLRIMAGE MErge [modifiers]: view d1...dn; Model Ent
d1...dn <CR>
```

Modifiers

Modifiers are listed alphabetically by level. A description of each modifier follows.

ANALYTICAL: Specifies the ANALYTICAL engine of the UPDATE HLRIMAGE operation.

BLANK: All hidden lines are blanked. This is the default. It cannot be used with the DASH modifier.

DASH: Displays hidden lines as dashed lines. It cannot be used with the BLANK modifier. The default dash font is DASH. You can use DLEN and SLEN to modify the dash font.

DLEN: Specifies the dash length in absolute drawing units. The default dash length is 0.25 drawing units.

SLEN: Specifies the inter-dash spacing in absolute drawing units. The default dash spacing is 0.25 drawing units.

DISSOCIATE: Converts an associate HLR Image into a non-associate image. This is applicable to the CAMU environment only.

FACETEDENGINE: Specifies the Faceted engine that approximates the object for HLR by tessellating it according to the Quality factor specified. The speed or quality of HLR operation will depend on the value specified for QUALITYFACTOR.

QUALITYFACTOR: Specifies a quality factor ranging from a low of 0 to a high of 100 and is inversely proportional to the speed of HLR image output. This modifier is available only while using the FACETEDENGINE modifier.

INTERFERENCE: Enables processing designed to minimize errors that may arise from the presence of interfering objects in the image. The use of INTERFERENCE is not generally recommended as it may degrade performance.

INTERSECT: Displays the intersection curves between objects. It automatically enables INTERFERENCE. This modifier involves a Boolean operation between the interfering objects and may take significant time for very complex and large assemblies and parts.

ILAYER: Specifies the layer to which the HLR intersection curves will belong. If the layer is not specified, the active layer value is used.

ITOL: Specifies the tolerance to be used for the intersection process. If the tolerance is not specified, the default tolerance value is used. This submodifier cannot be used along with the FACETEDENGINE modifier.

MERGE: Enables selection of objects in an HLRed view and performs HLR on the selected objects during UPDATE HLRIMAGE. You may select either the Faceted or the Analytical engine for this operation. If neither is specified, the engine chosen for the last HLR operation will be used.

ALL: Updates all objects visible in the selected view.

REFINE: This modifier provides better quality results of an HLR image. REFINE cannot be used along with the FACETEDENGINE modifier.

REMOVETANEDGES: Enables suppression of all tangent edges for the selected objects.

REMOVETOPEDGES: Enables suppression of all topological edges for the selected objects.

SHOWTOPEDGES: Shows all topological edges for the selected objects. A topological edge of a solid is the one that exists between two surfaces that are analytically identical.

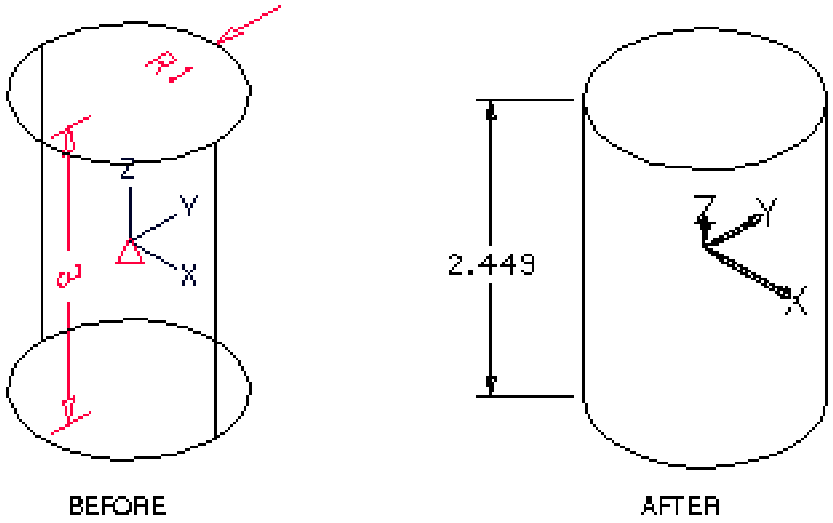
STATUS: Specifies the status of the selected HLR Image. That is it tell you whether the HLR image in the view is associative or nonassociative in nature.

SHOWTANEDGES: Shows all tangent edges for the selected objects. A tangent edge of a solid is one for which the tangent planes of the faces sharing the edge match at every point along the edge.

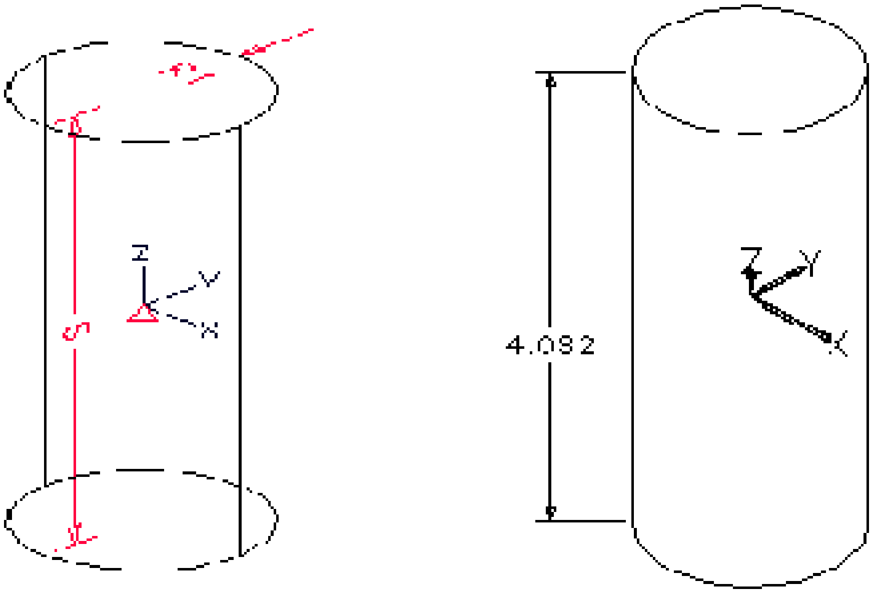
UPDATE: Enables to bring back the picture in sync with the model. When an entity in PART environment undergoes a parametric change or in CAMU a remote part corresponding to a component is modified, these dimensions are automatically updated according to the changes made in the entity.

Example

The first figure shows a cylinder having radius of 1 and height 3 generated in the Parametric mode. When the HIDE OBJECT command is applied in the explicit mode and then dimension is attached to one of the Limb's, the figure gets modified as shown below.



The cylinder is then modified in the parametric mode from 3 to 5. After using the UPDATE HLRIMAGE modifier the dimensions of the previously generated HLR Image is modified to reflect the changes as shown below.



RESTRICTIONS

Simultaneous display of exploded and un-exploded graphics on one drawing sheet is achievable. When the view parts are unblanked, the nonassociative surrogates may be out of synch with the viewed (unhidden) wireframe.

HLR picking and MARK CHANGE will not work for nonassociative HLR images. Important features that are supported through curve associativity (such as preservation of dimensions during updates) will not be supported.

- UNHIDE OBJECT, applied to specific entities in a view, may be used to remove them from the object list for subsequent executions of the UPDATE HLRIMAGE command. However, you cannot incrementally add entities to the object list. An expanded object list can only be accomplished by re-executing HIDE OBJECT.
- UPDATE HLRIMAGE provides the same automatic updating of surrogate curves as HIDE OBJECT.
- Unlike HIDE OBJECT, UPDATE HLRIMAGE provides multi-view HLR with the possibility of including a different set of objects in each view. If the same set of objects is to be hidden in each view, then HIDE OBJECT provides a superior performance.
- In order to assure that associativity in the HLRIMAGE is properly mapped between the parametric and explicit environments, the following line should be in the user's`.caddsrc` file:

  ```
  setenv CADDSASSOC_UID yes
  ```
- HIDE OBJECT can only maintain associativity to a viewed part when specific internal data exists in the viewed part. All CADDS design parts, except older CADDS parts, automatically contain this information. Such older parts have to be activated and filed as design parts in order to ensure subsequent associativity with respect to HLR.

Appendix A

User Aids

This appendix explains about the following user aids that are available with Appearance and Hidden Line Removal:

- Dynamic Menus
- Online Documentation

The file naming conventions presented here may be used at CADDS level only.

Dynamic Menus

Two dynamic menus are available: CVAEC.VIS.DMENU and CVAEC.GEN.DMENU.PHA**.**

CVAEC.VIS.DMENU contains the commands in the AEC Hidden Line Removal product. To activate the Main Menu, enter

```
#n# ACTIVATE DMENU CVAEC.VIS.DMENU
```

In addition, a tab is displayed for each command.

CVAEC.GEN.DMENU.PHA provides patternhatch selections for inserting patterns and textures with the command INSERT PHATCH. The menus icons illustrate the patterns. For more information on INSERT PHATCH, see the online command reference documentation. To activate the menu, enter

```
#n# ACTIVATE DMENU CVAEC.GEN.DMENU.PHA
```

Patternhatch commands and their modifiers are included in the key file called KEYFILE.CVAEC.GEN.PHATCH.P**.**

Online Documentation

There is online documentation for every Explicit Modeling command. You can access this information with the exclamation mark (!) or the question mark (?).

To display a list of verbs you can use with your version of CADDS, enter a question mark (?).

```
#01#?
```

When you press RETURN, you are returned to the CADDS prompt.

To find out which nouns are available with a particular verb, enter the verb followed by a question mark.

```
#01# GENERATE?
```

A list of nouns that can be used with the GENERATE command is displayed. When you press RETURN, you are returned to the system prompt.

To open up a Viewfile window that includes the description of a command and its modifiers, enter an exclamation point followed by the verb and the noun.

Your command line might look like this:

```
#01#!GENERATE DETAIL
```

Appendix B

System Messages

This appendix lists system messages in alphabetical order according to the first letter in each message. Following each message is an explanation, where necessary, and a list of the commands to which that message applies.

A

```
At least three selections are required to define a PWINDOW.
```

The command is not executed.

EDIT APPEARANCE

C

```
CANNOT RESOLVE PARAMETERS FOR MIPTR n. THIS ERROR USUALLY
OCCURS BECAUSE AN NFIG STOCK NUMBER CANNOT BE FOUND IN THE
PARAMETER FILE.
```

You can use the HIGHLIGHT PLINE MIPTR command to identify the entity (see the *Piping Reference*).

GENERATE DETAIL

```
CLEARANCE SHAPES REPEATED
```

More than one CLEARANCE header keyword was input.

COMPILE PROCEDURE

```
COMPILED FILE CANNOT BE STORED
```

Check disk space. There probably is no more room on the disk.

COMPILE PROCEDURE

```
COMPILED FILE OVERFLOW
```

There is too much in the procedure. Try to simplify by using fewer shapes, fewer parameters, etc.

COMPILE PROCEDURE

```
COMPILED PROCEDURE filename DOES NOT EXIST
```

The procedure file `filename` is not on the disk. COMPILE PROCEDURE is automatically executed to compile the procedure file.

GENERATE DETAIL

```
<n> COMPONENT(S) BEING PREPARED.
```

This is the first line of a two or three line message. The command is executing. Each nodal figure is counted as one component.

CREATE RENDERING
MERGE RENDERING

```
n COMPONENTS BEING SOLVED
```

This message appears before hiding begins. Each nodal figure is counted as one component. The message helps to indicate how long the process will take. If processing time exceeds one minute, a series of ten dots is output, each indicating that approximately 10% of the required processing has been completed.

GENERATE DETAIL
GENERATE DLINE
HIDE PIPE

```
CURRENT LINE BEING PROCESSED: <delimited text>
```

Identifies the line containing the problem.

CREATE RENDERING

D

`DIVISION BY ZERO IS INVALID.`

EDIT APPEARANCE

`DUPLICATE KEYWORD — SAME SHAPE`

The same keyword is found more than once in a shape definition.

COMPILE PROCEDURE

E

`ERROR IN ORIENTATION, X & Y AXES CANNOT BE PARALLEL.`

CREATE RENDERING

`ERROR IN THE PROCEDURE FILE <procedure name>`

This is the first line of a two or more line message. The command is executed for all the other nodal figures but not for this procedure.

CREATE RENDERING

`EXPRESSION IS INCOMPLETE.`

CREATE RENDERING

`EXPRESSION SIZE GREATER THAN 30 CHARACTERS`

Any expression KEYWORD=DATA is limited to 30 characters.

COMPILE PROCEDURE

`Extra selection ignored.`

You have exceeded the maximum number of selections required to specify a WINDOW or PWINDOW or to identify an interval on a line. The last selection is ignored and processing continues.

EDIT APPEARANCE

F

```
FIRST WORD NOT “SHAPE”.
```

Enter SHAPE= <text> in the procedure file.

EDIT APPEARANCE

I

```
ILLEGAL EXPRESSION
```

A keyword expression contains some illegal character(s).

COMPILE PROCEDURE

```
ILLEGAL UNITS SPECIFICATION
```

Unit must be MM, IN, M, FT, CM, KM, or MI.

COMPILE PROCEDURE

```
INVALID SHAPE NAME.
```

EDIT APPEARANCE

M

```
MISSING “=”
```

EDIT APPEARANCE

N

```
NEW LINE NOT A VALID LINE.
```

This usually means that a plus sign is needed in column 1 to indicate a continuation. Each line in the procedure file must start with one of the following:

- Plus sign (+), indicating a continuation
- Asterisk (*), indicating a comment
- “SHAPE”, indicating a new shape line

Edit the procedure file and reexecute the command.

EDIT APPEARANCE

```
NO CLEARANCE SHAPES INPUT
```

A CLEARANCE keyword is input, but no shapes are entered following it. This results in a null clearance procedure; the item associated with it is ignored during clearance checking. The procedure is still compiled.

COMPILE PROCEDURE

```
NO ENTITY WAS PROCESSED
```

CREATE RENDERING

```
NO SHAPES INPUT
```

A file is being compiled that has no shape definitions. It contains only comment lines.

COMPILE PROCEDURE

```
NO VIEW(S) DEFINED.
```

The command is not executed. This indicates that you have not issued the command SELECT MODE DRAW VIEW: `view` d1. Do this, then reexecute the command.

GENERATE DLINE

```
NOT A VALID PHRASE.
```

EDIT APPEARANCE

```
NOT ENOUGH KEYWORDS TO DEFINE SHAPE
```

See Table C-2, “Keywords and Their Meanings” for a listing of required keywords.

COMPILE PROCEDURE

```
NUMBER OR VARIABLE NOT FOUND WHEN ONE WAS EXPECTED.
```

EDIT APPEARANCE

O

```
OPERATOR NOT FOUND WHEN ONE WAS EXPECTED.
```

Line needs an arithmetic operator.

EDIT APPEARANCE

P

```
PARAMETERS FOR PROCEDURE <filename> CANNOT BE RETRIEVED.
```

This indicates that the parameter file or PLIST property does not exist or cannot be read. This may occur if the stock number cannot be found in the parameter file. Check the header lines of the parameter file. If they are not in the correct format, the file cannot be read.

GENERATE DETAIL

```
PARAMETER REQUIRED BUT NOT SET.
```

One of the necessary parameters is missing.

EDIT APPEARANCE

```
<n> PERCENT PROCESSED.
```

The command is executing. When many components are being processed, this message indicates the percent processed at 20% intervals.

CREATE RENDERING
MERGE RENDERING

```
PIPE HAS MORE THAN ONE STRAIGHT SEGMENT
```

You can only detail pipes (Nlines) that consist of one straight segment between connect nodes (Cnodes). Any invalid pipe is ignored and processing continues.

GENERATE DETAIL

```
PIPE DOES NOT HAVE OUTER DIAMETER FOR GRAPHICS
REPRESENTATIONS
```

The OUTERDIAM property is missing so detail graphics cannot be generated for that particular pipe. Processing still continues.

GENERATE DETAIL

`PROBLEM READING INPUT FILE(S).`

The command is aborted. The procedure or parameter file is unreadable and must be re-created or restored from archive. Reexecute the command.

CREATE RENDERING

`PROCESS COMPLETED, GENERATING GRAPHICS.`

MERGE RENDERING

R

`RECOMPILING PROCEDURE` <filename>

The procedure file filename is being compiled automatically by the COMPILE PROCEDURE command. If accompanied by the message `COMPILED PROCEDURE` filename `DOES NOT EXIST`, the procedure was not on the disk. If this message alone is given, the version on the disk needed updating. During the compilation, system messages generated by COMPILE PROCEDURE may appear.

GENERATE DETAIL

S

`SHAPE TYPE NOT ALLOWED FOR CLEARANCE`

A shape was entered that is not recognized as a clearance shape. The procedure file is still compiled. If there is a CLEARANCE section, the message only pertains to shapes in the CLEARANCE section.

COMPILE PROCEDURE

`SQUARE ROOT OF A NEGATIVE NUMBER IS INVALID.`

EDIT APPEARANCE

`STACK OVERFLOW.`

The expression was nested too deep.

EDIT APPEARANCE

```
STACK UNDERFLOW.
```

Unbalanced nested parentheses.

EDIT APPEARANCE

```
SYNTAX
```

The line is output with a caret pointing to the problem.

COMPILE PROCEDURE

T

```
THE <n> HIGHLIGHTED (RED) ENTITIES REMAIN ON THEIR ORIGINAL
LAYERS
```

The command is executed. The creation layer of a Model entity is not changed.

CREATE RENDERING

```
The interval specified is not a linear interval [LINTERVAL].
```

The command is not executed. The portion of the string or nodal line you have identified contains a vertex and cannot be processed. Execute the command again.

EDIT APPEARANCE

```
THE PROCEDURE FILE DOES NOT EXIST FOR <text>
```

Nodal figures require a procedure file. The command is executed for all nodal figures with a procedure file. Enter a nodal figure name. Reexecute the command.

CREATE RENDERING

```
TOO MANY LINES
```

This indicates that too many vertices are selected for the polygon window.

HIDE PIPE
UNHIDE PIPE

```
Two selections are required to define a WINDOW.
```

EDIT APPEARANCE

The command is not executed.

U

```
UNRECOGNIZABLE KEYWORD
```

The keyword indicated is invalid.

COMPILE PROCEDURE

```
UNRECOGNIZABLE SHAPE
```

You entered an incorrect shape keyword and shape name.

COMPILE PROCEDURE

V

`VARIABLE <`parameter name`> REFERENCED BUT NOT SET.`

The command is executed, but detailed graphics are not drawn for the specified fitting. The procedure file lists <parameter name> as a parameter for a fitting, but the parameter file does not have a value for <parameter name>. Therefore, no detailed graphics are generated for this fitting. Check the parameter file, modify the procedure file, or add the PLIST property.

GENERATE DLINE

```
VARIABLE "T" REFERENCED BUT NOT SET
```

A valid value for the variable in the PLIST property is required.

CREATE RENDERING

W

```
WARNING: MIRRORED NFIG MIPTR n may require new stock number.
```

You have mirrored an item that is not identical when it is reversed (for example, an eccentric reducer). It now needs a different stock number.

GENERATE DETAIL

```
WARNING: SELECT PPARAMETERS PAFILE IS MISSING
```

The default parameter file you selected with the SELECT PPARAMETERS command does not exist. Items that require this file for checking will not be checked.

GENERATE DETAIL

Appendix C

Basic Procedure File Shapes

This appendix lists and illustrates the basic shapes procedure file shapes supported by COMPILE PROCEDURE, GENERATE DETAIL, CHECK CLEARANCE (see the *Reports and Analyses Reference*), and GENERATE SURFACE. This appendix also provides information on defining shapes in a procedure file.

- Defining Shapes

Defining Shapes

Each shape definition in the procedure begins with the keyword SHAPE= followed by the shape name and expressions for dimensions, location, and orientation. The expressions have the format KEYWORD=DATA. Data can include the following:

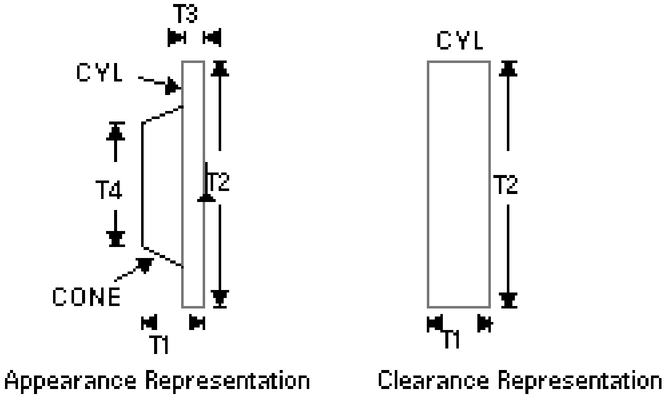
- Parameters or variables (T1,T2,T3...)
- Component scales (SX,SY,SZ)
- Constants (integer/real)
- Simple mathematical operators (+,y,*,/,^)
- Functions (SQRT,SIN,COS,TAN)
- Parentheses () to define subexpressions (nesting is permitted)

Use commas to separate expressions. Continue a definition to the next line with a plus sign (+) in column 1. (The maximum is two continuation lines.) Use an asterisk (*) in column 1 for comment lines.

The following is an example of a procedure. The procedure defines the representations in the figure on the next page.

```
*
SHAPE=CONE,D1=T4,D2=1.2*T4,L=T1-T3
SHAPE=CYL,L=T3,D1=T2
*
CLEARANCE
SHAPE=CYL,L=T1,D1=T2
*
```

The first shape defined is a cone with a smaller diameter of T4, a larger diameter of 1.2 times T4, and a length of T1 — T3; the second shape is a cylinder with a length of T3 and a diameter of T2. A clearance representation for the component is defined under the keyword CLEARANCE: a cylinder with a length of T1 and a diameter of T2.



The dimensions of the shapes are expressed in terms of parameters (such as T1, T2, T3, T4), so that you can represent many sizes of a component with the same procedure. The values for the parameters are taken either from a parameter file or from a PLIST property.

You may use the same shapes for detailing, checking clearance, and generating surfaces. Clearance checking is performed with the CHECK CLEARANCE command (for more information, see the *Reports and Analyses Reference*). Shapes used only for clearance checking are put after the CLEARANCE keyword. If you do not use the CLEARANCE keyword, the appearance representation is used for checking. To eliminate checking, input the CLEARANCE keyword with no shape names listed after it.

Table C-1 lists the allowable shapes, the names you use for them in the procedure file, and their required keywords (dimensional data). Table C-2, “Keywords and Their Meanings”lists the keywords and their meanings

Table C-1 Supported Shapes

Shape Name	Shape	Keywords
BOX	Box	L, H, W, and eccentric keywords if tilted in local space
CONE	Cone	L, D1, D2
CYL	Cylinder	L, D1
ECON	Eccentric Cone	L, D1, D2, EDX, EDY, EDZ
ECT1	Circular throat, rectangular heel, rectangular cross-section elbow	W, H, W2, H2
ELLH	Elliptical Head	R, RMIN

Table C-1 Supported Shapes

Shape Name	Shape	Keywords
ERT1	Rectangular throat, circular heel, rectangular cross-section elbow	W, H, W2, H2
GRRT	General rectangular to rectangular transition	L2, W, H, D1, YOFF1,ZOFF1
GRTR	General round-to-round transition	L2,ZOFF1, D1, D2, YOFF1
GTRN	General rectangular to rectangular transition	L2, W, H, W2, H2,YOFF1, ZOFF1
HEMI	Hemispherical Head	R
MITR	Mitre Elbow, rectangular	W, H, W21, H21
MIT1	Mitre Elbow, round	D1, D21
MIT2	2 Cut Mitre Elbow, round	D1, D2
MIT3	3 Cut Mitre Elbow, round	D1, D2
MIT4	4 Cut Mitre Elbow, round	D1, D2
PCDF or TRAN	Pyramidal diffuser or transition rectangularto rectangular	L, W, H, W2, H23
PECD or SETR	Single pyramidal, eccentric center, diffuser or transition	L, W, H, W2, H23
PEDF or DETR	Double eccentric pyramidal diffuser, transition rectangular to rectangular	L, W, H, W2, H23
RCEL	Elbow, rectangular	W, H, W21, H21
RRDF	Diffuser, rectangular toround	L, W, H, D1
RREL	Elbow, rectangular toround	W, H, D1
SPH	Sphere	R
TOR	Toroid	D1, D21
WYDR	Wye-dovetail, rectangular	W, H, W21, H21

1 Optional keywords

2 Optional when using internal nodal lines

3 When using CREATE RENDERING, MERGE RENDERING, or SHADE MODEL, the keywords L, D1, H, D2, and HE must be used

Table C-2 Keywords and Their Meanings

Keywords	Meanings
SHAPE	Specifies shape name.
X,Y,Z	Define coordinate to locate shape origin relative to component origin. (Default: X0Y0Z0). Not necessary for 1.First shape when its origin coincides with component origin. 2.Subsequent shapes when their origins coincide with the destination of the previous shape.

Table C-2 Keywords and Their Meanings

Keywords	Meanings
DX, DY, DZ	Define a coordinate into which to rotate the positive x-axis of the shape. Default to DX=1 if no components of coordinate are specified. Unspecified components default to zero.
EDX, EDY,EDZ	Same as DX, DY, DZ except that these define a coordinate into which to rotate the positive y-axis of the shape. Defaults to EDY=1 if no components of coordinate are specified. Unspecified components default to zero.
L	Length of cylinder, cone, box, etc.
D1	Diameter of cylinder, initial diameter for cones, toroids, and elbows. Starting width if HE is used.
D2	End diameter for cones, toroids, and elbows. Ending width if HE is used.
H	Height of box cross section, or rectangular section at inlet of GTRN, GRRT, etc.
H2	Height of rectangular section at outlet of GTRN, GRRT, etc.
HE	Height of rectangular section at outlet of GTRN, GRRT, etc.
W	Width of rectangular section at inlet of GTRN, GRRT, etc.
W2	Width of rectangular section at outlet of GTRN, GRRT, etc.
R	Radius of sphere, hemi-head, and 1/2 major axis of elliptical head.
RMIN	One-half minor axis of elliptical head.
YOFF	Defines an offset in the positive y-axis of the shape by which the downstream end of the shape will be moved with respect to the upstream end.
ZOFF	Defines an offset in the positive z-axis of the shape by which the downstream end of the shape will be moved with respect to the upstream end.

Table C-3, “Basic Shape Information” includes the following shape information.

1. The name of the shape.
2. How the shape is handled for clearance checking — whether it is the shape itself or an automatic approximation that is used for checking. The shapes listed with Same are checked directly (the clearance representation is the same as the shape itself); other shapes use the indicated approximations. The clearance representations for RREL and RRDF include two shapes (CHECK CLEARANCE checks to see if either shape causes interference). ELLH and HEMI are also approximated with two shapes, but no interference is reported unless both shapes interfere. Shapes approximated with a single box use the larger height and width dimensions in the parameter listing. ERT1 and ECT1 shapes are not supported by the CHECK CLEARANCE command.
3. Whether or not the shape is used by the CONSTRUCT EQUIPMENT command. (For more information on CONSTRUCT EQUIPMENT, see the *Visualization/Preparation Reference*).
4. Whether or not the shape follows the path of the internal Nline. Shapes that follow the path of the internal Nline are approximated with cylinders or boxes,

one for each Nline segment; the diameter of the cylinders or the width and height of the boxes are taken from the corresponding dimensions of the shape.

5. For shapes that follow the Nline path, the number of segments that are appropriate/recommended for the internal Nline. For MITR and MIT1, the number of segments depends on the numbers of cuts you want. For RCEL, RREL, and WYDR, 20 segments are recommended to produce an acceptably smooth curve.

6. Whether or not the eccentricity keywords (EDX, EDY, and EDZ) apply.

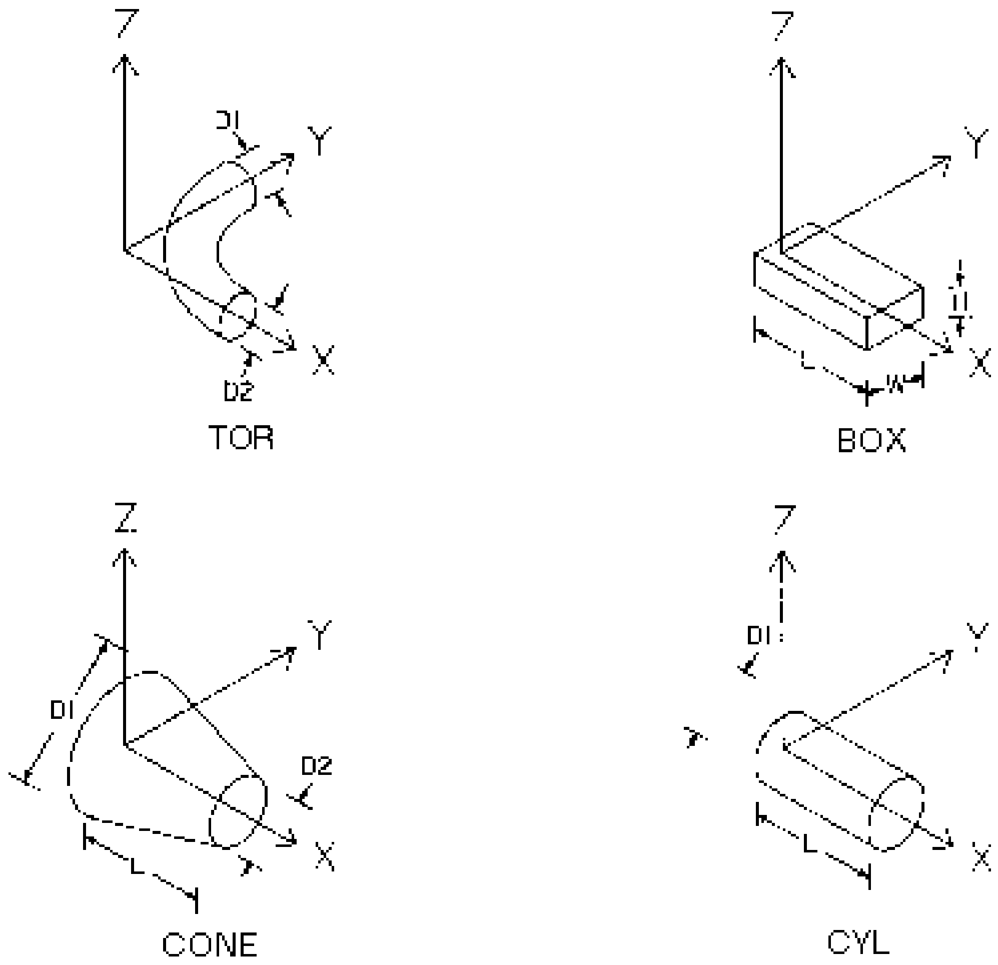
Table C-3 Basic Shape Information

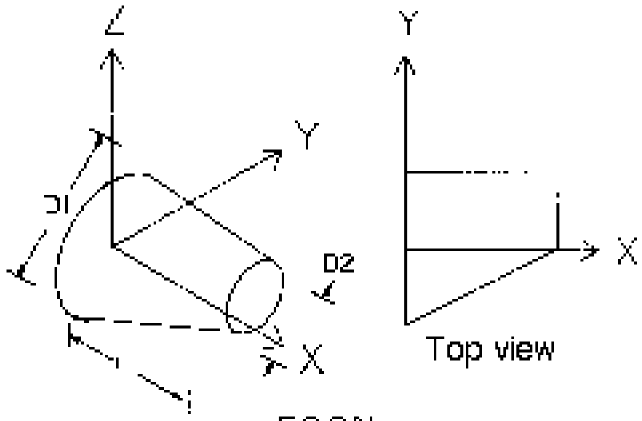
Shape Name	Clearance Shape(s)	Valid for CON.EQUIP	Follows NlinePath	# Segments Recommended	Keywords EDX,EDY,EDZ
BOX	Same	yes	no space		If tilted in local
CONE	Same	yes	no		
ECON	Cylinder	yes	no		yes
ECT1	not supported	no	yes		
CYL	Same	yes	no		
ELLH	Intersection of cylinder and sphere	yes	no		
ERT1	not supported	no	yes		
GRRT	GTRN	no	no		
GRTR	Cylinder	no	no		
GTRN	GTRN	no	no		
HEMI	Same	yes	no		
MITR	Boxes	no	yes	2-20	
MIT1	Cylinders	no	yes	2-50	
MIT2	Cylinders	no	yes	2-50	
MIT3	Cylinders	no	yes	2-50	
MIT4	Cylinders	no	yes	2-50	
PCDF	Box	no	no		yes
PECD	Box	no	no		yes
PEDF	Box	no	no		yes
RCEL	Boxes	no	yes	20	
RRDF	Box and cylinder	no	no		yes
RREL	MITR1and MITR	no	yes	20	
SPH	Same	yes	no		
TOR	Cylinders	no	yes		
WYDR	2 MITRS	no	yes	20	yes

Please note: When the location keywords (X,Y,Z) are not used, the first shape is placed at the component origin (X0Y0Z0). Each subsequent shape is placed at the end of the previous shape. When the orientation keywords (DX,DY,DZ,EDX,EDY,EDZ) are not used, the first shape is oriented along the component positive x-axis; each subsequent shape is oriented in the same direction as the previous shape. When any location or orientation keywords are used, the defaults used for omitted keywords are relative to the component's origin and x-axis (as for the first shape). The defaults are not relative to the previous shape.

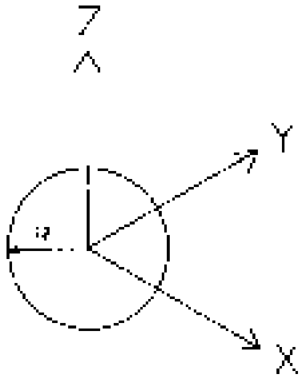
Before putting shapes together to build a component, you must know the location of their origins and main axes. Most of their geometry is predefined, but for a few shapes dimensional expressions are required. For example, the values for D1 and D2 determine which end of the cone is larger and whether it is truncated.

The basic shapes are illustrated in the following illustration.

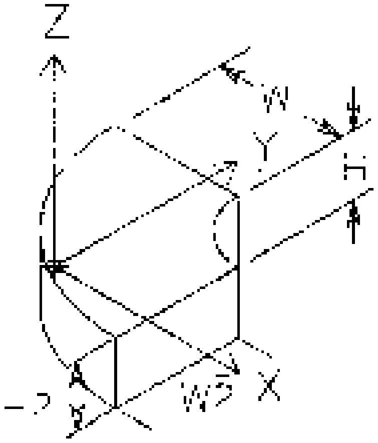




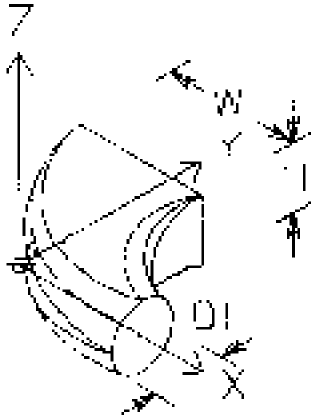
ECON



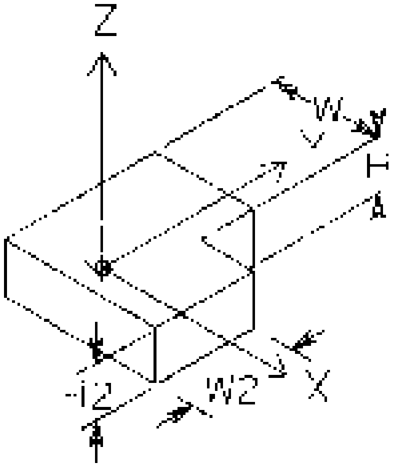
SPH



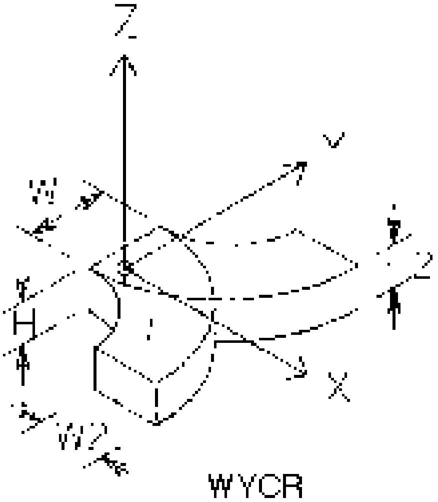
RCEL



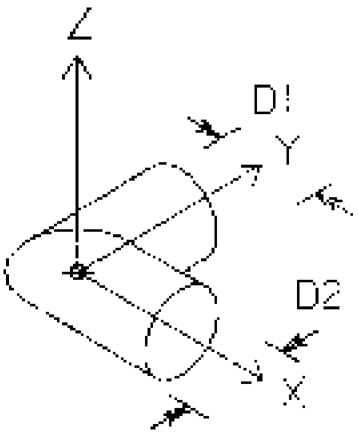
-



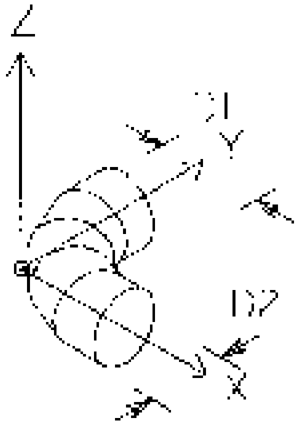
MITR



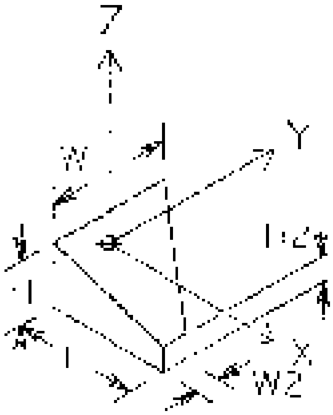
WYCR



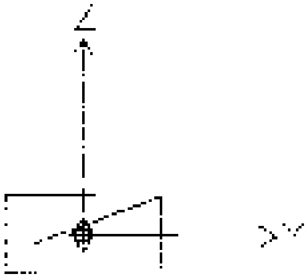
MIT1



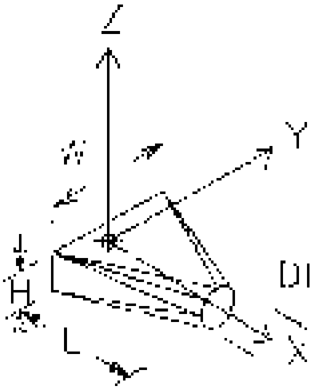
MIT4



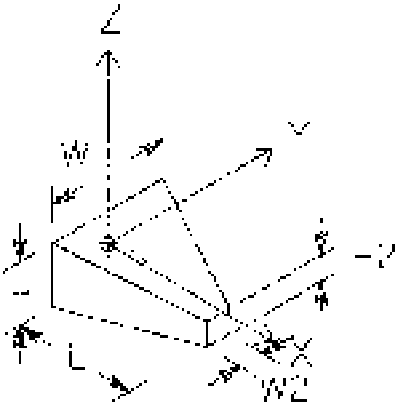
PEDF



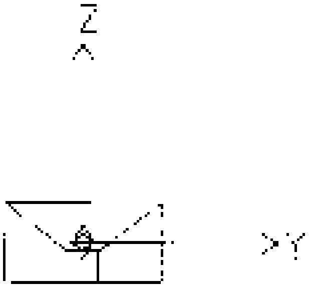
Right View



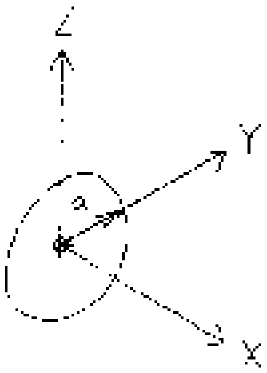
RRDF



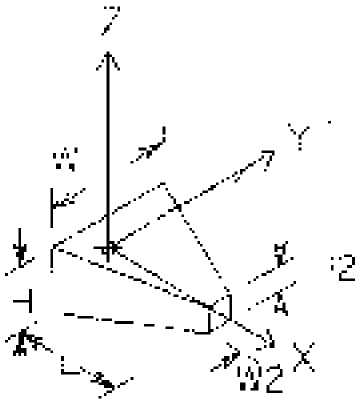
PECD



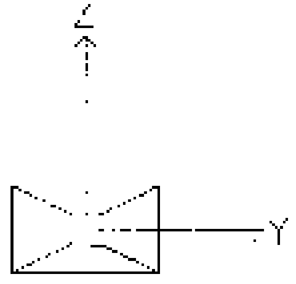
Right View



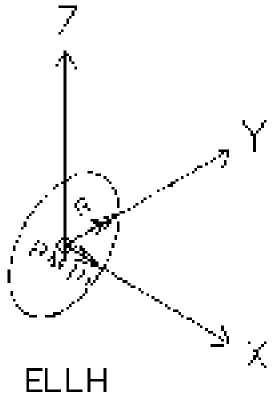
HEMI



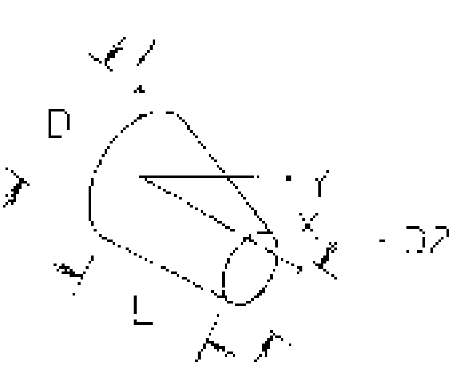
PCDF



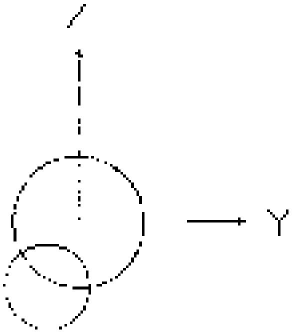
Right View



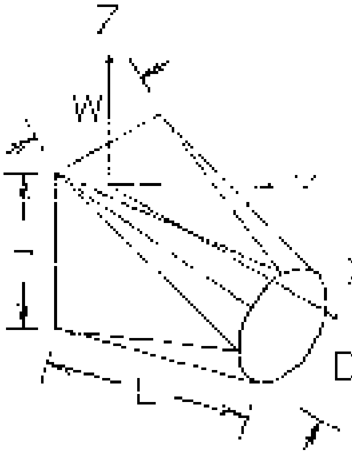
ELLH



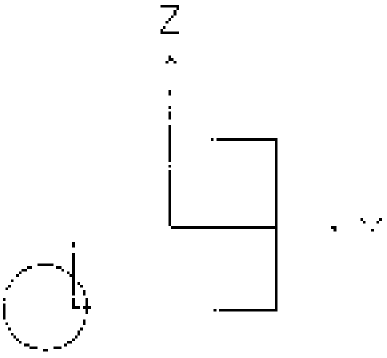
GRTR



Right View



GRRT



Right View

CONSTRUCT EQUIPMENT and PREPARE EQUIPMENT Commands

You can automatically create a procedure file with the CONSTRUCT EQUIPMENT and PREPARE EQUIPMENT commands. CONSTRUCT EQUIPMENT lets you see the shapes as you dimension, orient, and position them. PREPARE EQUIPMENT automatically writes and compiles the procedure file.

However, since many shapes (including those for ducts) cannot be used with CONSTRUCT EQUIPMENT, you will need to know how to develop procedures and how to modify existing ones. For more information on CONSTRUCT EQUIPMENT and PREPARE EQUIPMENT, see the *Visualization/Preparation Reference*.

Defining the Dimensions, Location, and Orientation of Shapes

Before using a procedure to generate detail graphics or check clearances, you must define the dimensions, locations, and orientations of the shapes. Define these parameters directly by constants or by values obtained from a parameter file, a PLIST (parameter list) property, or component scales.

Use the parameter file for components such as pipeline fittings that vary by size. Use the PLIST for unique items such as equipment. Indicate the parameters in expressions with T1, T2, T3, and so on. You may combine component scales with T's.

Dimensions

The component's dimensions in a three-dimensional model are given by its X, Y, and Z scales. You can dimension shapes relative to these scales by using the keywords SX, SY, and SZ in place of T's (for example, L=SX, D1=SY). For the extents of the detail graphics to coincide with the model graphics, the totals of the dimensional parameters for each shape along the x-, y-, and z- axes must equal the component's respective X, Y, and Z scales (the sum of the T's along the x-axis equals the x-scale of the component). If the X, Y, and Z scales coincide with the parameters (for example, T1 = X, T2 = Y, T3 = Z), parameters do not have to be used; SX, SY, and SZ keywords can be used instead.

Please note: The unit of the data (parameters and constants) used in expressions defaults to inches. To specify other units, include a unit statement line at the top of the procedure file or parameter file. The format of the units statement is

```
UNITS=XX.
```

where *XX* is equal to MM, CM, IN, FT, M, KM, or MI. This line must be left justified, with no spaces.

Location and Orientation

To represent a component with a set of shapes, first locate and orient the individual shapes in the local coordinate system (or space) of the component. The term local space defines a coordinate system relative to the origin and axes of the component, without regard to its location in the larger (or global) context of a three-dimensional model. The set of shapes as a whole is then positioned and rotated with the component when it is used in a three-dimensional model.

By default, the origin of the first shape is placed at the component's origin. The x-axis of the shape is aligned with the component's x-axis, and each subsequent shape is placed and oriented relative to the previous shape (in a chaining fashion). Thus, each successive shape begins where the previous shape left off and continues in the same direction until you use location and orientation keywords to define another direction. Note that the process begins over again for clearance shapes.

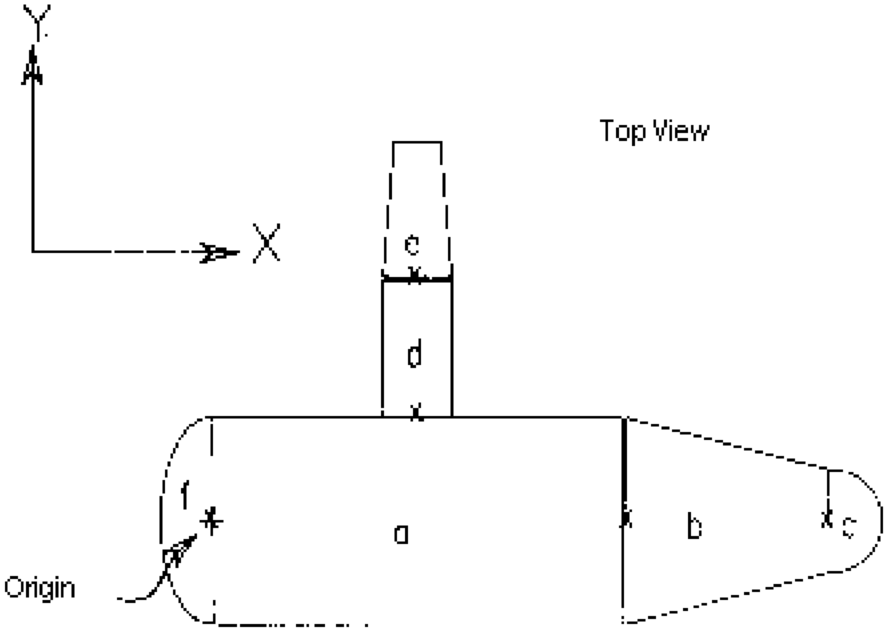
You may locate and orient any shape independent of other shapes with the keywords X, Y, Z, DX, DY, DZ, EDX, EDY, EDZ (refer to the keyword table in this description). These keywords specify a coordinate (location or direction vector) in the local space of the component. You can use variables as well as real numbers in orientation expressions (for example, DZ = 1 + T1/2). Independent location and orientation of a shape changes the default for subsequent chaining. If you do not give a value for an orientation keyword, the value defaults to 0. For example, if you specify DZ = 1 and do not specify DX and DY, the result is X0, Y0, Z1.

The DX, DY, DZ, EDX, EDY, and EDZ keywords allow a method of rotation referred to as the implicit form or the vector-to-vector method. The point of rotation is the origin of the shape. In the first case the x-axis (vector) is rotated until it becomes coincident with the axis (vector) specified by the point you input. For example, DY=1 specifies the point X0Y1Z0 (unspecified coordinates default to zero). This input causes the shape's x-axis to be rotated 90 degrees counterclockwise (toward the y-axis) so that it runs through the point. In the second case, the y-axis (vector) is rotated into the point specified by EDX, EDY and EDZ. For example, EDY=1 and EDZ=1 specify the point X0 Y1Z1, into which the shape's y-axis is rotated 45 degrees counterclockwise.

Please note: When using both sets of orientation keywords, do not rotate the x- and y-axis of the shape such that they are no longer perpendicular.

Example

The following illustration and key show how a set of shapes is put together.



The origin of each individual shape is marked with an x.

The illustration is defined by this procedure:

```
(a) SHAPE=CYL,L=T1,D1=T2
(b) SHAPE=CONE,L=T1*.5,D1=T2,D2=T2*.5
(c) SHAPE=HEMI,R=T2*.25
(d) SHAPE=CYL,L=T1*.333,D1=T2*.333,X=T1*.5,Y=T2*.5,DY=1
(e) SHAPE=ECON,L=T1*.333,D1=T2*.333,D2=T2*.222
(f) SHAPE=ELLH,R=T2*.5,RMIN=T2*.25,X=0,Y=0,Z=0,DX=y1
```

The following key explains how the procedure relates to the illustration.

a-c

A cylinder, cone, and hemisphere are automatically located in succession along the positive x-axis, beginning at the origin of the figure (+).

d

The origin of this cylinder is specified relative to +; its orientation is specified by DY = y1 (so that it is not chained to e).

e

This eccentric cone takes its orientation and location from d.

f

The final shape, an elliptical head, has its origin specified by X = 0, Y = 0, Z = 0 (you do not need to specify coordinates that do not change); its origin is specified by DX = y1 (so that it is not chained to e).

Index