

Understanding CVNC™

CADD5® 5 Revision 6.1.2

DOC38503-006

Copyright © 2001 Parametric Technology Corporation. All Rights Reserved.

User documentation from Parametric Technology Corporation (PTC) is subject to copyright laws of the United States and other countries and is provided under a license agreement, which restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed user the right to make copies in printed form of PTC user documentation provided on software or documentation media, but only for internal, noncommercial use by the licensed user in accordance with the license agreement under which the applicable software and documentation are licensed. Any copy made hereunder shall include the Parametric Technology Corporation copyright notice and any other proprietary notice provided by PTC. User documentation may not be disclosed, transferred, or modified without the prior written consent of PTC and no authorization is granted to make copies for such purposes.

Information described in this document 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. UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

Registered Trademarks of Parametric Technology Corporation or a Subsidiary

Advanced Surface Design, CADD5, CADDShade, Computervision, Computervision Services, Electronic Product Definition, EPD, HARNESSDESIGN, Info*Engine, InPart, MEDUSA, Optegra, Parametric Technology, Parametric Technology Corporation, Pro/ENGINEER, Pro/HELP, Pro/INTRALINK, Pro/MECHANICA, Pro/TOOLKIT, PTC, PT/Products, Windchill, InPart logo, and PTC logo.

Trademarks of Parametric Technology Corporation or a Subsidiary

3DPAINT, Associative Topology Bus, Behavioral Modeler, BOMBOT, CDRS, CounterPart, CV, CVact, CVaec, CVdesign, CV-DORS, CVMAC, CVNC, CVToolmaker, DesignSuite, DIMENSION III, DIVISION, DVSAFEWORK, DVS, e-Series, EDE, e/ENGINEER, Electrical Design Entry, Expert Machinist, Expert Toolmaker, Flexible Engineering, *i*-Series, ICEM, Import Data Doctor, Information for Innovation, ISSM, MEDEA, ModelCHECK, NC Builder, Nitidus, PARTBOT, PartSpeak, Pro/ANIMATE, Pro/ASSEMBLY, Pro/CABLING, Pro/CASTING, Pro/CDT, Pro/CMM, Pro/COMPOSITE, Pro/CONVERT, Pro/DATA for PDGS, Pro/DESIGNER, Pro/DESKTOP, Pro/DETAIL, Pro/DIAGRAM, Pro/DIEFACE, Pro/DRAW, Pro/ECAD, Pro/ENGINE, Pro/FEATURE, Pro/FEM-POST, Pro/FLY-THROUGH, Pro/HARNESS-MFG, Pro/INTERFACE for CADD5, Pro/INTERFACE for CATIA, Pro/LANGUAGE, Pro/LEGACY, Pro/LIBRARYACCESS, Pro/MESH, Pro/Model.View, Pro/MOLDESIGN, Pro/NC-ADVANCED, Pro/NC-CHECK, Pro/NC-MILL, Pro/NC-SHEETMETAL, Pro/NC-TURN, Pro/NC-WEDM, Pro/NC-Wire EDM, Pro/NCPOST, Pro/NETWORK ANIMATOR, Pro/NOTEBOOK, Pro/PDM, Pro/PHOTORENDER, Pro/PHOTORENDER TEXTURE LIBRARY, Pro/PIPING, Pro/PLASTIC ADVISOR, Pro/PLOT, Pro/POWER DESIGN, Pro/PROCESS, Pro/REPORT, Pro/REVIEW, Pro/SCAN-TOOLS, Pro/SHEETMETAL, Pro/SURFACE, Pro/VERIFY, Pro/Web.Link, Pro/Web.Publish, Pro/WELDING, Product Structure Navigator, PTC *i*-Series, Shaping Innovation, Shrinkwrap, The Product Development Company, Virtual Design Environment, Windchill e-Series, CV-Computervision logo, DIVISION logo, and ICEM logo.

Third-Party Trademarks

Oracle is a registered trademark of Oracle Corporation. Windows and Windows NT are registered trademarks of Microsoft Corporation. Java and all Java based marks are trademarks or registered trademarks of Sun Microsystems, Inc. CATIA is a registered trademark of Dassault Systems. PDGS is a registered trademark of Ford Motor Company. SAP and R/3 are registered trademarks of SAP AG Germany. FLEX/m is a registered trademark of GLOBEtrouter Software, Inc. VisTools library is copyrighted software of Visual Kinematics, Inc. (VKI) containing confidential trade secret information belonging to VKI. HOOPS graphics system is a proprietary software product of, and copyrighted by, Tech Soft America, Inc. All other brand or product names are trademarks or registered trademarks of their respective holders.

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) or DFARS 227.7202-1(a) and 227.7202-3(a), and are provided to the 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 or Commercial Computer Software-Restricted Rights at FAR 52.227-19, as applicable.

Parametric Technology Corporation, 140 Kendrick Street, Needham, MA 02494-2714

8 January 2001

Table of Contents

Preface

| | |
|--|------|
| Related Documents _____ | xiii |
| Book Conventions _____ | xiv |
| Window Managers and the User Interface _____ | xv |
| Online User Documentation _____ | xv |
| Online Command Help _____ | xvi |
| Printing Documentation _____ | xvi |
| Resources and Services _____ | xvii |
| Documentation Comments _____ | xvii |

Introduction to CVNC

| | |
|---|------|
| What Is CVNC? _____ | 1-2 |
| Where CVNC Fits in the NC Process _____ | 1-4 |
| CVNC Concepts _____ | 1-6 |
| CVNC: Provides a Full Set of NC Functions _____ | 1-7 |
| CVNC-M2 _____ | 1-7 |
| CVNC-M3 _____ | 1-8 |
| CVNC-M5 _____ | 1-9 |
| CVNC-T2 _____ | 1-10 |
| CVNC-P2 _____ | 1-10 |
| CVNC Batch _____ | 1-10 |
| CVNC: Combines Language and Graphic Technologies _____ | 1-11 |

| | |
|---|------|
| Part Programming Methods _____ | 1-11 |
| The CVNC Environment _____ | 1-12 |
| CVNC Commands _____ | 1-12 |
| Information Hierarchy _____ | 1-13 |
| CVNC: Integrated with CADD5 _____ | 1-15 |
| Referencing CADD5 Geometry _____ | 1-15 |
| Changes to Geometry _____ | 1-16 |
| CVNC: Works with a Range of Integrated Applications Software _____ | 1-18 |
| CVNC: Provides a Sophisticated User Interface _____ | 1-19 |
| CVNC: Allows User Customization _____ | 1-22 |
| Command and Execute Files _____ | 1-22 |
| Status Display _____ | 1-22 |
| CVMAC Macros _____ | 1-22 |
| Pass-Through Statements _____ | 1-23 |
| CVNC Information _____ | 1-24 |
| Understanding CVNC (this book) _____ | 1-24 |
| CVNC System User Guide and Menu Reference _____ | 1-24 |
| CVNC Applications User Guides _____ | 1-24 |
| CVNC Editor Guide _____ | 1-25 |
| CVNC Command Reference Information _____ | 1-25 |
| CVNC System Variables Guide _____ | 1-25 |
| CVNC Command and Interface Cross-Reference _____ | 1-25 |
| CVNC Work Examples _____ | 1-25 |
| Customizing CVNC _____ | 1-25 |
| CVNC Master Index _____ | 1-25 |
| Introducing the Sample Part _____ | 1-26 |

CVNC Features

| | |
|----------------------------|-----|
| User Interface _____ | 2-3 |
| CADD5 CVNC Interface _____ | 2-3 |
| Icons and Menus _____ | 2-5 |
| Editing Facilities _____ | 2-5 |

| | |
|--|------|
| Property Sheets _____ | 2-8 |
| Tooling Management _____ | 2-9 |
| Online Documentation _____ | 2-9 |
| Documentation Window _____ | 2-10 |
| List of Commands _____ | 2-11 |
| List of Commands Starting with 'S' _____ | 2-11 |
| CADD5 Access _____ | 2-12 |
| Job Control File (JCF) _____ | 2-13 |
| How Are Job Control Files Created _____ | 2-13 |
| System Variables _____ | 2-15 |
| CVNC System Variables _____ | 2-15 |
| Displaying Variables Status _____ | 2-15 |
| Entity Groups _____ | 2-16 |
| CVNC Editor _____ | 2-17 |
| Editor Facilities _____ | 2-17 |
| Updating Graphics _____ | 2-17 |
| CVNC Star Buffers _____ | 2-17 |
| Editor Commands _____ | 2-17 |
| Tool and Tool Path Display _____ | 2-19 |
| Displaying Tools _____ | 2-19 |
| Displaying Tool Paths _____ | 2-20 |
| Tool Path Verification _____ | 2-22 |
| NC Tool Path Verifier Features _____ | 2-22 |
| Command Decomposition _____ | 2-24 |
| Command and Execute Files _____ | 2-25 |
| CVMAC Macros _____ | 2-26 |
| Output _____ | 2-27 |
| CLFile Output _____ | 2-28 |
| APT Output _____ | 2-29 |
| COMPACT II Output _____ | 2-30 |
| CVNC Application Summaries _____ | 2-31 |
| CVNC-M2 _____ | 2-31 |

| | |
|---|------|
| CVNC-M3_____ | 2-31 |
| CVNC-M5_____ | 2-32 |
| CVNC-T2_____ | 2-32 |
| CVNC-P2 _____ | 2-33 |
| CVNC-M2_____ | 2-34 |
| Tool Shape Definition _____ | 2-34 |
| Configuration _____ | 2-35 |
| Discrete Motion _____ | 2-35 |
| Depth Control_____ | 2-35 |
| Stock _____ | 2-35 |
| Boundary Profiling _____ | 2-36 |
| Pocketing with Islands _____ | 2-36 |
| Lace Pocketing with Islands_____ | 2-38 |
| Hole Processing _____ | 2-38 |
| CVNC-M3_____ | 2-39 |
| Integration with CVNC-M2 _____ | 2-39 |
| Straight Line Machining _____ | 2-39 |
| Parametric Machining _____ | 2-40 |
| Control Curve Machining _____ | 2-41 |
| Constant Z-Height Profile Finishing _____ | 2-41 |
| Uncut Area Marking _____ | 2-43 |
| 3-D Projection of 2-D Tool Paths _____ | 2-46 |
| Surface Lists _____ | 2-46 |
| Containment _____ | 2-47 |
| Surface Intersection Machining _____ | 2-47 |
| Collision Checking _____ | 2-48 |
| Approach and Retract Control _____ | 2-48 |
| Scallop Height and Stepover Control _____ | 2-49 |
| Surface Extensions_____ | 2-49 |
| 7-Parameter Tools _____ | 2-49 |
| Three-dimensional Profiling_____ | 2-49 |
| Material Modeling for Roughing_____ | 2-50 |
| CVNC-M5_____ | 2-51 |

| | |
|---|------|
| Integration with CVNC-M2 and M3 _____ | 2-51 |
| Gouge Avoidance _____ | 2-51 |
| Scallop Height and Stepover Control _____ | 2-52 |
| Swarfcutting _____ | 2-52 |
| Profiling _____ | 2-52 |
| Fixed or Variable Lead and Slew Control _____ | 2-54 |
| 4-Axis Parametric Machining _____ | 2-54 |
| CVNC-T2 _____ | 2-55 |
| Rough Turning, Boring, and Facing _____ | 2-55 |
| Boundary Profiling _____ | 2-56 |
| Discrete Motion _____ | 2-56 |
| Threading _____ | 2-56 |
| Grooving _____ | 2-57 |
| Configuration _____ | 2-57 |
| Tool Shape Definition _____ | 2-57 |
| Axis Mirroring and Transformation _____ | 2-58 |
| CVNC-P2 _____ | 2-59 |
| Machine and Turret Definition _____ | 2-59 |
| Punch Definition _____ | 2-59 |
| Variable Angle Nibbling and Punching _____ | 2-60 |
| Sorting and Optimizing Punches _____ | 2-61 |
| Area Clearance _____ | 2-62 |
| Configurable Output _____ | 2-62 |
| Automatic Search for Optimum Punch _____ | 2-62 |
| Repeating a Punching Operations Block _____ | 2-63 |
| CVNC Batch _____ | 2-64 |
| Creating a JCF _____ | 2-64 |
| Executing a JCF _____ | 2-64 |
| Viewing Results _____ | 2-65 |

The Numerical Control Solution

| | |
|--|-----|
| Creating a CAM-Ready CADDs Model _____ | 3-2 |
|--|-----|

| | |
|---|------|
| Designing the Mounting Bracket _____ | 3-3 |
| Design and Manufacturing Worked Together _____ | 3-4 |
| Layering Scheme _____ | 3-4 |
| Functional Geometry _____ | 3-5 |
| Views _____ | 3-5 |
| Creating the JCF _____ | 3-8 |
| How Is the JCF Created _____ | 3-8 |
| JCF Contents _____ | 3-8 |
| System Commands in the JCF _____ | 3-9 |
| Application Commands in the JCF _____ | 3-9 |
| Job Setup: Machine Configuration _____ | 3-10 |
| Configuring the Machine _____ | 3-10 |
| Job Setup: Defining Tools _____ | 3-12 |
| Defining Tools _____ | 3-12 |
| Job Setup: Generating Output _____ | 3-13 |
| Specifying Output Format _____ | 3-13 |
| Operation Setup _____ | 3-15 |
| Selecting Tools - CHGTOOL _____ | 3-15 |
| Selecting the Active Construction Plane - INDEX _____ | 3-15 |
| Grouping Entities - NCGROUP _____ | 3-16 |
| Defining Operation Parameters _____ | 3-16 |
| Application-specific Parameters _____ | 3-18 |
| Tool Motion Commands _____ | 3-19 |
| Cutting Motion for CVNC-M2 _____ | 3-19 |
| Cutting Motion for CVNC-M3 _____ | 3-19 |
| Cutting Motion for CVNC-M5 _____ | 3-20 |
| Cutting Motion for CVNC-T2 _____ | 3-20 |
| Cutting Motion for CVNC-P2 _____ | 3-20 |
| Creating the JCF for the Mounting Bracket _____ | 3-21 |
| Operation 1 (JCF OPR 1) _____ | 3-21 |
| Operation 2 (JCF OPR 2) _____ | 3-24 |
| Operation 3 (JCF OPR 3) _____ | 3-26 |
| Generating Output _____ | 3-33 |

| | |
|---|------|
| Generating Output for the Mounting Bracket _____ | 3-34 |
| CLFile for Opr 1 - Bottom-Mill _____ | 3-34 |
| CLFile for Opr 2 - Top-Drill _____ | 3-35 |
| CLFile for Opr 2 - Top-Drill _____ | 3-36 |
| CLFile for Opr 3 - Top-Mill _____ | 3-37 |
| Postprocessing _____ | 3-42 |
| CVGPll _____ | 3-42 |
| Configuration Tool _____ | 3-42 |
| Postprocessor _____ | 3-43 |
| Custom Postprocessors _____ | 3-43 |
| Postprocessing Output for the Mounting Bracket _____ | 3-44 |
| CVGPll Configuration Tool Main Menu _____ | 3-44 |
| Printing the Processed CLFile to the Terminal with mill2p _____ | 3-45 |
| Distributing Data _____ | 3-46 |
| Paper Tape _____ | 3-46 |
| Direct Numerical Control _____ | 3-46 |

Preface

Understanding CVNC provides an overview of the CVNC family of products. This book introduces CVNC and explains how this product fits into the complete numerical control solution. It is written for managers, CAD/CAM system administrators, and part programmers.

This book traces a sample part (a mounting bracket used on the space shuttle) from part design through manufacture.

For information on training, reporting software problems, and other product-related issues, consult your system manager.

Related Documents

The following documents may be helpful as you use *Understanding CVNC*:

- *CVNC System User Guide and Menu Reference*
- *CVNC Editor Guide*
- *Customizing CVNC*
- Command reference and user guide(s) for your application
- *CVNC Master Index*
- *CVNC Command and Interface Cross Reference*
- *CVNC System Variables Guide*
- *CVNC Work Examples*

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 | <code>cvaec.hd.data.param</code> <code>tar -xvf /dev/rst0</code> | Enter the text in a CADDs text field or on any command line. |
| System output | <code>Binary transfer complete.</code> | Indicates system responses in the CADDs text window or on any command line. |
| Variable in user input | <code>tar -cvf /dev/rst0 filename</code> | Replace the variable with an appropriate substitute; for example, replace filename with an actual file name. |
| Variable in text | 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 | <i>n</i> | Supply an integer for the <i>n</i> . |
| Real number | <i>x</i> | Supply a real number for the <i>x</i> . |
| # | # <code>mkdir /cdrom</code> | Indicates the root (superuser) prompt on command lines. |
| % | % <code>rlogin remote_system_name -l root</code> | Indicates the C shell prompt on command lines. |
| \$ | \$ <code>rlogin remote_system_name -l root</code> | Indicates the Bourne shell prompt on command lines. |

Window Managers and the User Interface

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

Look and Feel of User Interface Elements

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

Online User Documentation

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

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

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

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

From an HTML Browser:

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

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

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

From the Documentation CD-ROM:

1. Mount the documentation CD-ROM.
2. Point your browser to:
CDROM_mount_point/html/doc/mainmenu.html (UNIX)
CDROM_Drive:\html\doc\mainmenu.html (Windows NT)

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 NT)

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 in the following ways:

- Send comments electronically to `doc-webhelp@ptc.com`.
- Fill out and mail the PTC Documentation Survey located in the *PTC Customer Service Guide*.

Introduction to CVNC

This chapter provides an overview of CVNC, which is designed to help manufacturing engineers and part programmers automate NC programming processes.

CVNC is integrated with CADD5 and CVMAC, and each CVNC application includes a state-of-the-art user interface and comprehensive user documentation.

This chapter describes the many ways CVNC can be customized and used, and introduces the sample part that is used throughout this document to illustrate the steps involved in the NC solution.

- What Is CVNC?
- Where CVNC Fits in the NC Process
- CVNC Concepts
- CVNC: Provides a Full Set of NC Functions
- CVNC: Combines Language and Graphic Technologies
- CVNC: Integrated with CADD5
- CVNC: Works with a Range of Integrated Applications Software
- CVNC: Provides a Sophisticated User Interface
- CVNC: Allows User Customization
- CVNC Information
- Introducing the Sample Part

What Is CVNC?

The CVNC family of products are used to automate NC programming processes.

CVNC supports most manufacturing devices (such as 2- and 2 1/2-axis mills/drills, 3-axis machining centers, and front or rear single or twin turret 2-axis turning machines) and production engineering needs.

CVNC greatly increases part programmer productivity by

- Eliminating tedious calculations
- Decreasing error possibilities
- Reducing the need for the part programmer to be familiar with the specific machine tool programming requirements

The CVNC family of products (see Figure 1-1, “CVNC Family of Products,” on page 1-3) includes

- CVNC-M2 for mill/drill and machining centers
- CVNC-M3 for advanced 3-axis surface machining, fully integrated with CVNC-M2
- CVNC-M5 for advanced 4- and 5-axis surface machining, fully integrated with CVNC-M2 and CVNC-M3
- CVNC-T2 for single and twin turret lathes
- CVNC-P2 for punching machines
- CVNC Batch for batch processing JCFs (job control files)

The CVNC Base System, which contains the CVNC core technology and environment, is included with all CVNC products. CVNC-M3 requires CVNC-M2. CVNC-M5 requires CVNC-M3 and CVNC-M2. CVNC Batch requires CVNC-M2, CVNC-M3 or CVNC-M5.

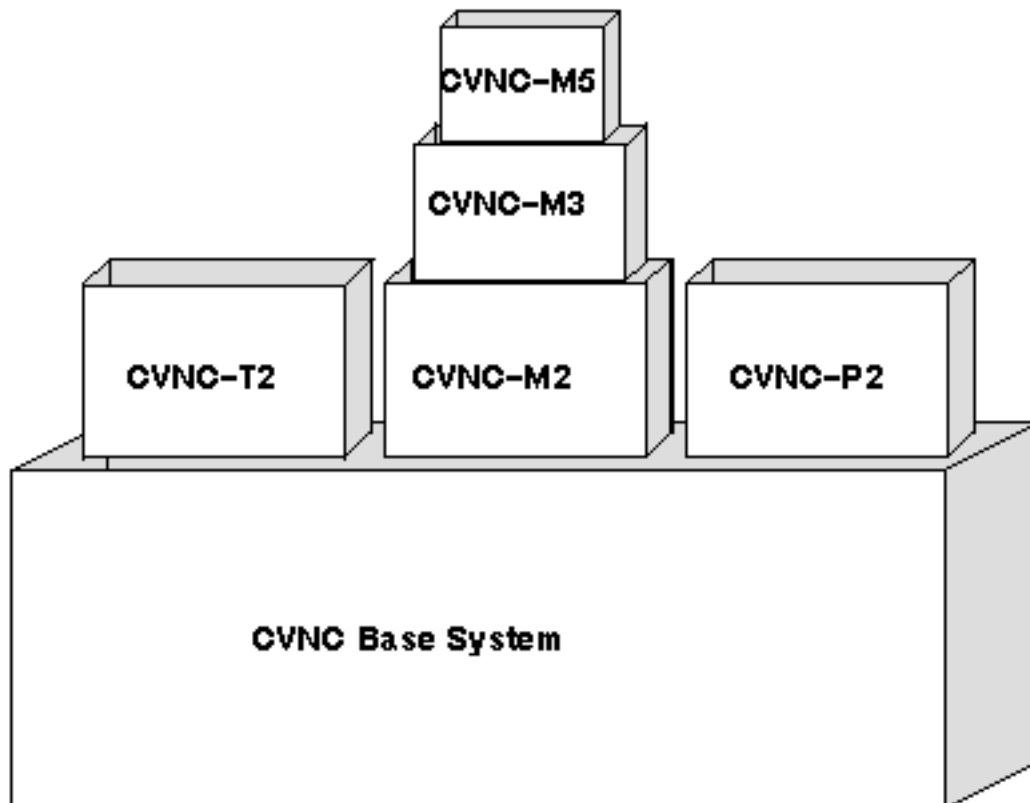
With CVNC, you use an easily understandable, shop-oriented language to program machining operations. CVNC is closely integrated with CADD5 and CVMAC.

CVNC provides

- Simple machine control and tool motion commands used interactively to create precise NC part programs.
- A task-oriented user interface that simplifies part programming.

- Customization with CVMAC macros that can be accessed with a single user-defined command.
- The ability to store machining instructions and descriptions. For example, you can store part programs to create a library of NC processes for machining different parts. You can also create a machine tool library that can be accessed by other NC processes.

Figure 1-1 CVNC Family of Products

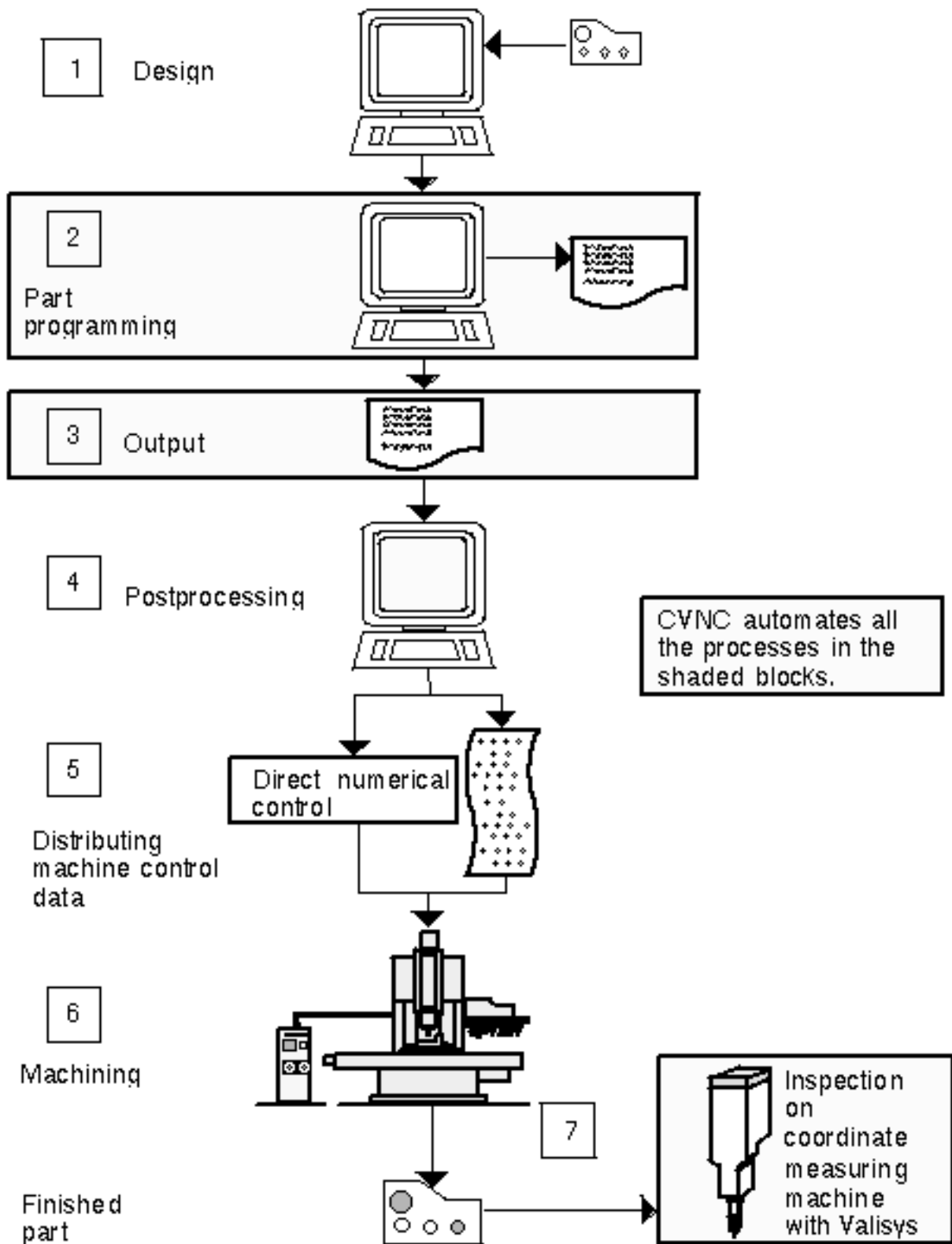


Where CVNC Fits in the NC Process

CVNC automates the numerical control part programming step of the manufacturing process. The following list and the figure on the next page show where CVNC fits in the manufacturing process:

1. A part is designed with CADDSS.
2. A JCF (Job Control File) part program is created with the appropriate CVNC product. For example, for a drilling operation, you would use CVNC-M2. CVNC uses the same database created with CADDSS, and retrieves all necessary coordinate information about the part from this database to calculate tool motion. CVNC part programs reference tagged entities in the CADDSS model.
3. CVNC generates an output file from the part program in an industry standard format (CLFile, APT, COMPACT II). The language choice depends on your output processor type.
4. Postprocessing software turns the output file into machine-readable data (MCD, machine control data). For example, if you are using postprocessor, CVGPPII, you will output a CLFile. CVGPPII translates the CLFile into MCD appropriate for the machine tool being used.
5. The MCD is sent to the machine electronically or via tape. (The CVNC tape utility is included with your CVNC software.)
6. The part is machined.
7. After machining, you can inspect the accuracy of the part with a coordinate measuring machine (CMM). CADDSS Valisys software creates a program for a coordinate measuring machine to drive a probe cluster around part features and generate a report.

Figure 1-2 CVNC: Part of the Big Picture



CVNC Concepts

CVNC concepts result in its benefits over conventional part programming and other NC software products. CVNC concepts are determined by users. CVNC users include manufacturing engineers and part programmers. Users need a full range of NC capabilities to support a variety of NC machines.

- CVNC provides a full set of NC functions.
Users need a system with a part programming language that is easy to learn, can be edited, can be saved and referred to offline, and corresponds to a graphic display.
- CVNC combines language and graphic technologies in a part programming environment that is easy for even a novice user.
Users need a system that works hand-in-hand with the CADDs database created for part design.
- CVNC is fully integrated with CADDs.
Users need a system that works with other software to take a part from design through quality assurance.
- CVNC works with a range of integrated applications software that together provide a complete NC solution.
Users need a system with a state-of-the-art interface that will satisfy programmer requirements at different levels.
- CVNC's sophisticated user interface guides a novice user through operations and allows an experienced user to increase productivity.
Users need a powerful and flexible system that can satisfy individual shop requirements.
- CVNC can be user-customized to make its power and application virtually limitless.

These concepts are described on the following pages.

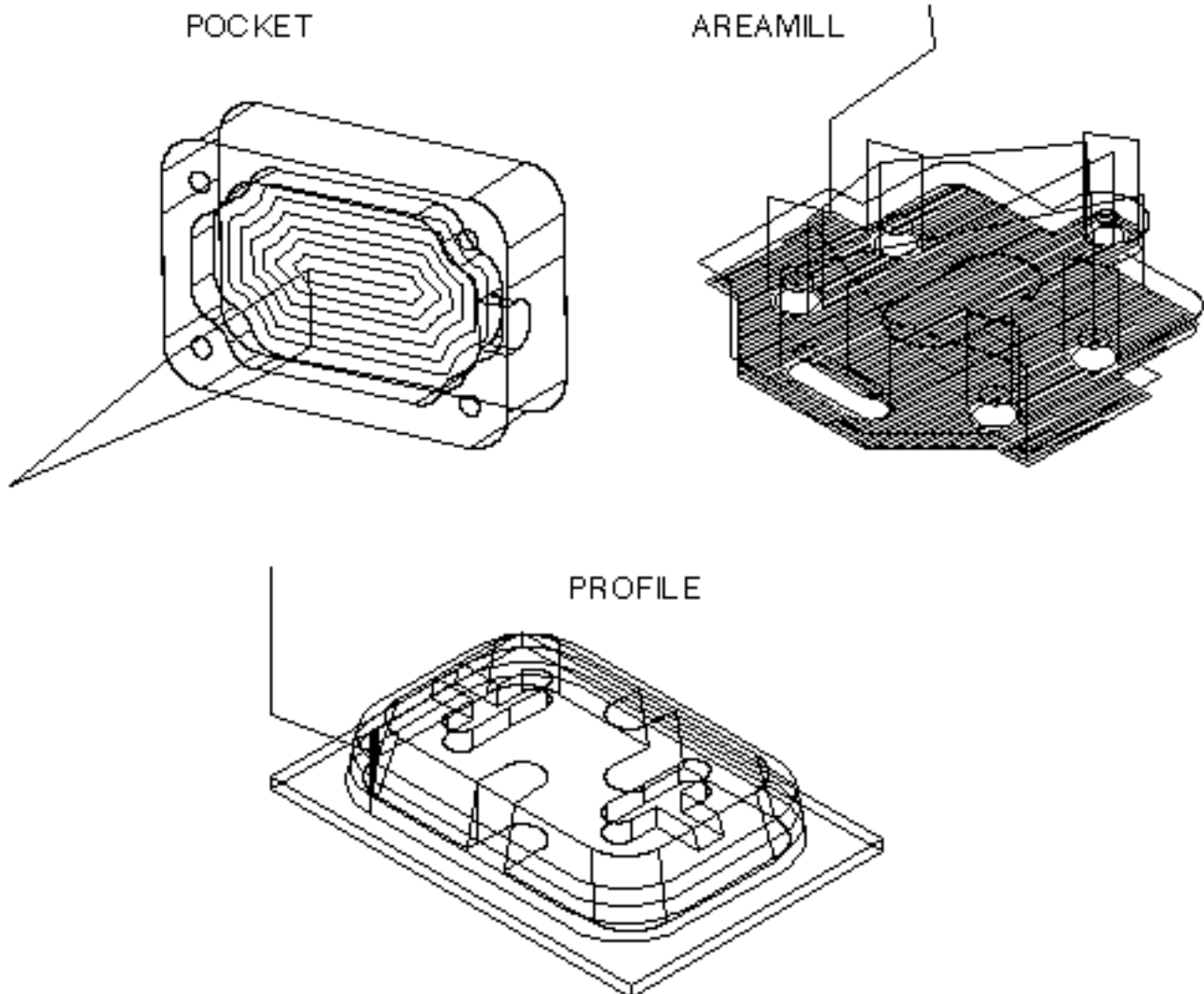
CVNC: Provides a Full Set of NC Functions

CVNC supports a large variety of manufacturing devices with its six applications: CVNC-M2, CVNC-M3, CVNC-M5, CVNC-T2, and CVNC-P2. Each application provides powerful NC functions. CVNC applications are described in more detail in Chapter 2, “CVNC Features”.

These selected features illustrate the broad range of CVNC functions.

CVNC-M2

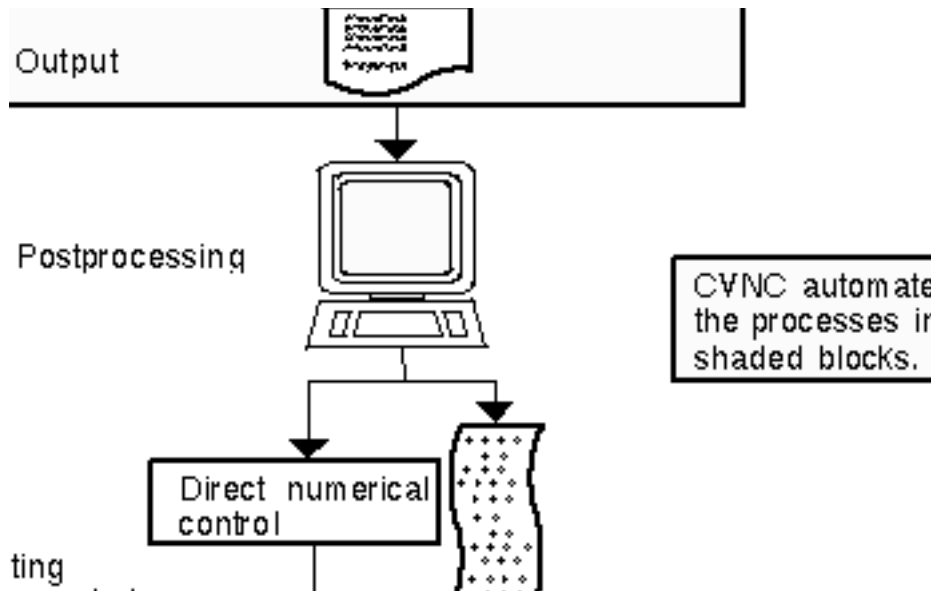
CVNC-M2 is used for 2 1/2-axis milling and drilling applications. It provides profiling, pocketing, area clearance by lace pocketing, drilling, boring, countersinking, tapping, and other NC functions.



CVNC-M3

CVNC-M3 provides all CVNC-M2 functionality, plus

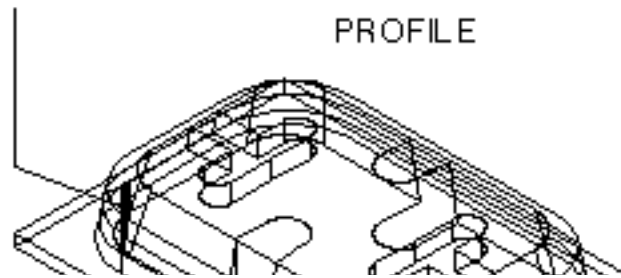
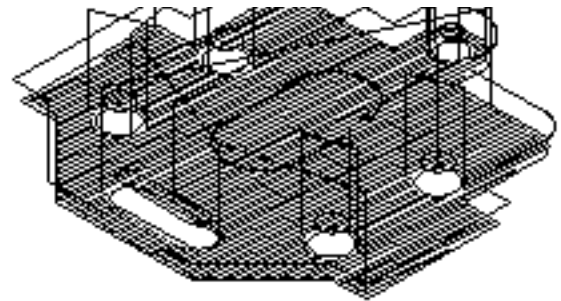
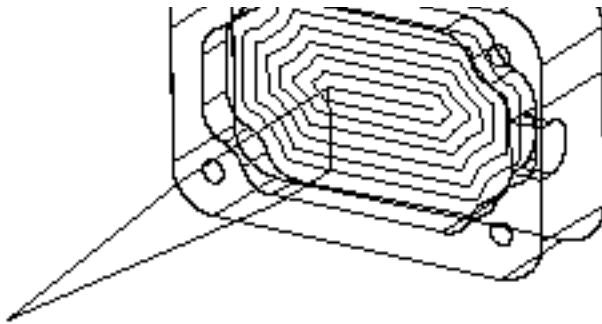
- Three-axis parametric, straight line, constant z-height, and control curve surface and surface intersection machining over multiple surfaces (including 7-parameter tool support)
- Collision checking
- Uncut area marking
- Gouge avoidance
- Three-dimensional contouring
- Generation and merging of containment boundaries (including 3-axis wireframe containment)
- Projection of two-dimensional tool paths onto three-dimensional surfaces
- Uphill machining
- Rough machining using material modeling
- Other NC functions



CVNC-M5

CVNC-M5 provides all CVNC-M2 and CVNC-M3 functionality, plus

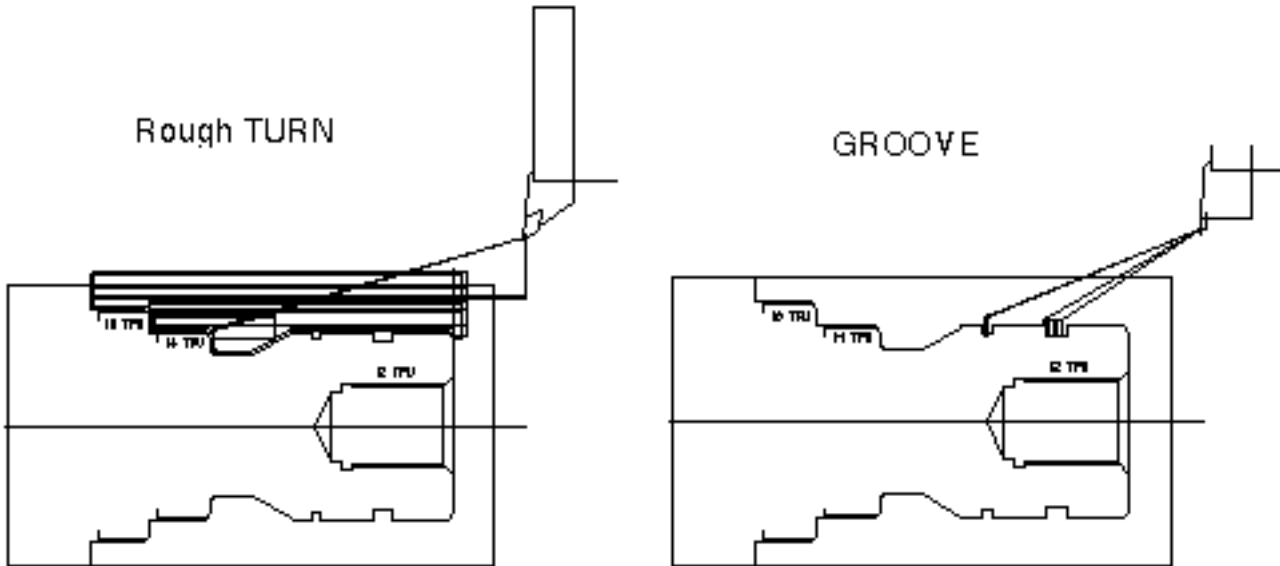
- 5-axis straight line, parametric and control curve surface machining with tool axis controls
- 5-axis surface intersection machining
- APT-7 parameter tool shape support
- Swarfcutting
- 5-axis profiling of curves and surfaces
- Collision and gouge avoidance
- Other NC functions



In addition, CVNC-M5 includes 4-axis (three linear and one rotational axes) parametric surface machining functions and 4-axis surface intersection machining (with tapered tool support).

CVNC-T2

CVNC-T2 provides 2-axis turning capabilities, including threading, rough turning, tapping, grooving, drilling, and other NC functions.



CVNC-P2

CVNC-P2 provides 2-axis punching capabilities, including punching, nibbling, sorting and optimizing, area clearance, and other NC functions.

CVNC Batch

CVNC Batch enables you to execute a JCF without a CADDSS session. The main advantage of doing this is that you need not wait for each command in the JCF to execute as soon as it is entered, as in CADDSS CVNC. You can enter all the commands in the JCF and then run the JCF at a convenient time.

You have to do the following in CVNC Batch

- Create a JCF in CADDSS CVNC
- Execute the JCF in CVNC Batch
- See the results in CADDSS CVNC

CVNC: Combines Language and Graphic Technologies

CVNC combines easy to use language and graphic technologies in an advanced part programming environment.

Part Programming Methods

Without CVNC, a part program must be written directly in machine control language, as shown in the figure below, Part Program in Machine Control Language, or in a processor language, such as APT or COMPACT II, as shown in Figure 1-4, “Part Program in APT,” on page 1-11.

Figure 1-3 Part Program in Machine Control Language

```
N0011X155.5$
N0012S1200$
N0013G00C-180.M03F0$
N0014X8.961Y5.5088M08$
N0015Z14.122$
N0016G01Z2.F0.5$
N0017G09Z1.5F8.5$
N0018X8.5408Y5.8061$
N0019G03X8.2677Y9.0177I+2.0408J+1.4439$
N0020G02X7.7114Y13.2427I+2.4743J+2.4749$
N0021G08$
N0022G01X7.7115Y13.2427Z.F80.5$
```

Figure 1-4 Part Program in APT

```
1  PARTNO 8888                                010
2  CLPRNT                                       012
3  NOPOST                                       014
4  SP      = POINT/ 0, 0, 0                    020
5  L1      = LINE/ 4, 0, 0, 4, 8, 0            030
6  PT      = POINT/ 4.0, 8.0, 0               040
7  L2      = LINE/ PT, ATANGL, 45             050
8  L3      = LINE/ 8, 12, 0, 12, 12, 0       060
9  L4      = LINE/ 14, 5, 0, 14, 10, 0       070
10 INTOL/ 0                                    080
11 OUTTOL/ .005                                090
12 CUTTER/ .25                                 100
13 SPINDL/ 2000, CL                            110
14
```

Machine control language is very cumbersome. It requires the part programmer to be fluent in the language for each individual machine tool and to make numerous calculations to properly position the tool. Processor languages are specialized for NC, but they are not written in an environment that interacts with graphics and allows editing during part programming.

With CVNC, a part program can be written in an advanced, easy to use programming language, as shown in the Figure 1-5 below.

Figure 1-5 Part Program in CVNC

```
[JCF CVNC MILL      6-20-92 11:16 CADDSS]
AUDIT ON
#>*
#>* SIDE 1 - OPERATION 1
OUTPUT CLFILE 'BOTTOM-MILL'
SELECT LAY 150
#>* ZPLANE TOP = 0
READ 'TOP' TOPPLANE
PLANE ZREF 0
PLANE ZAPPR 1.04 ZREF
PLANE ZCLEAR 1 ZREF
PLANE ZRETRACT .05 ZREF
```

The CVNC Environment

CVNC provides a simultaneous graphic display of tool path and tool motion and the part program text. Screen graphics always reflect the part program up to the current line position. The CVNC editor allows you to step back, make changes, and step forward, all within the part program. The graphic display updates as edits are made.

CVNC Commands

CVNC is driven by its command language. Although CVNC commands can be input via the CVNC interface, they are stored sequentially in a file called a Job Control File (JCF). The JCF

- Provides a permanent record of all operations
- Can be written and analyzed offline
- Can be shared, edited, and reexecuted
- Can be batch processed

CVNC commands allow you to specify job setup, operation parameters, and tool motion in a shop-oriented language easily understood. You will find the straightforward CVNC command language requires a short learning curve.

Information Hierarchy

The highest information level is the process plan. It identifies the operations necessary to manufacture a part.

Generally, one JCF (the next information level) is written for each machine setup. High-level CVNC commands can be broken down (decomposed) into low-level data for program fine-tuning.

For example, the AREAMILL lace pocketing command can be decomposed into low-level statements for each cut, as shown in Figure 1-6, “Decomposed Command Line,” on page 1-13.

You can use CVNC editing commands to step forward, edit, and back over decomposed command lines, just as you can to the original, high-level command.

The following figure illustrates the information hierarchy.

Figure 1-6 Decomposed Command Line

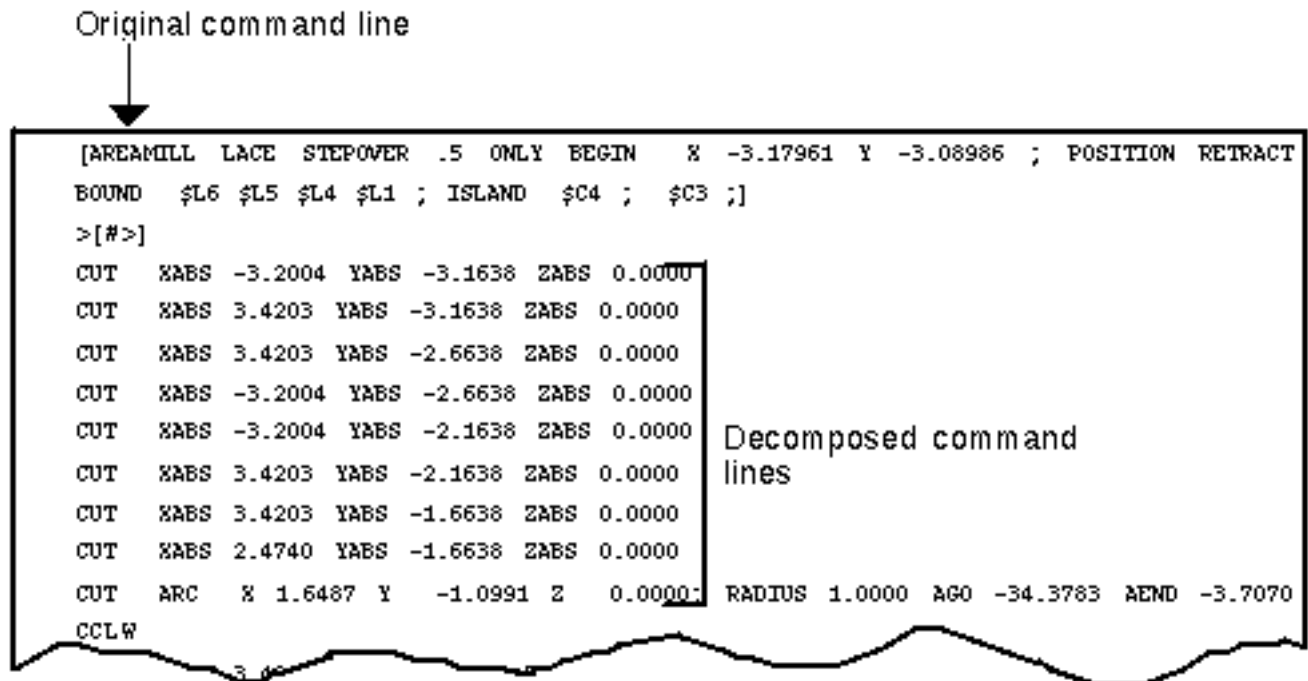
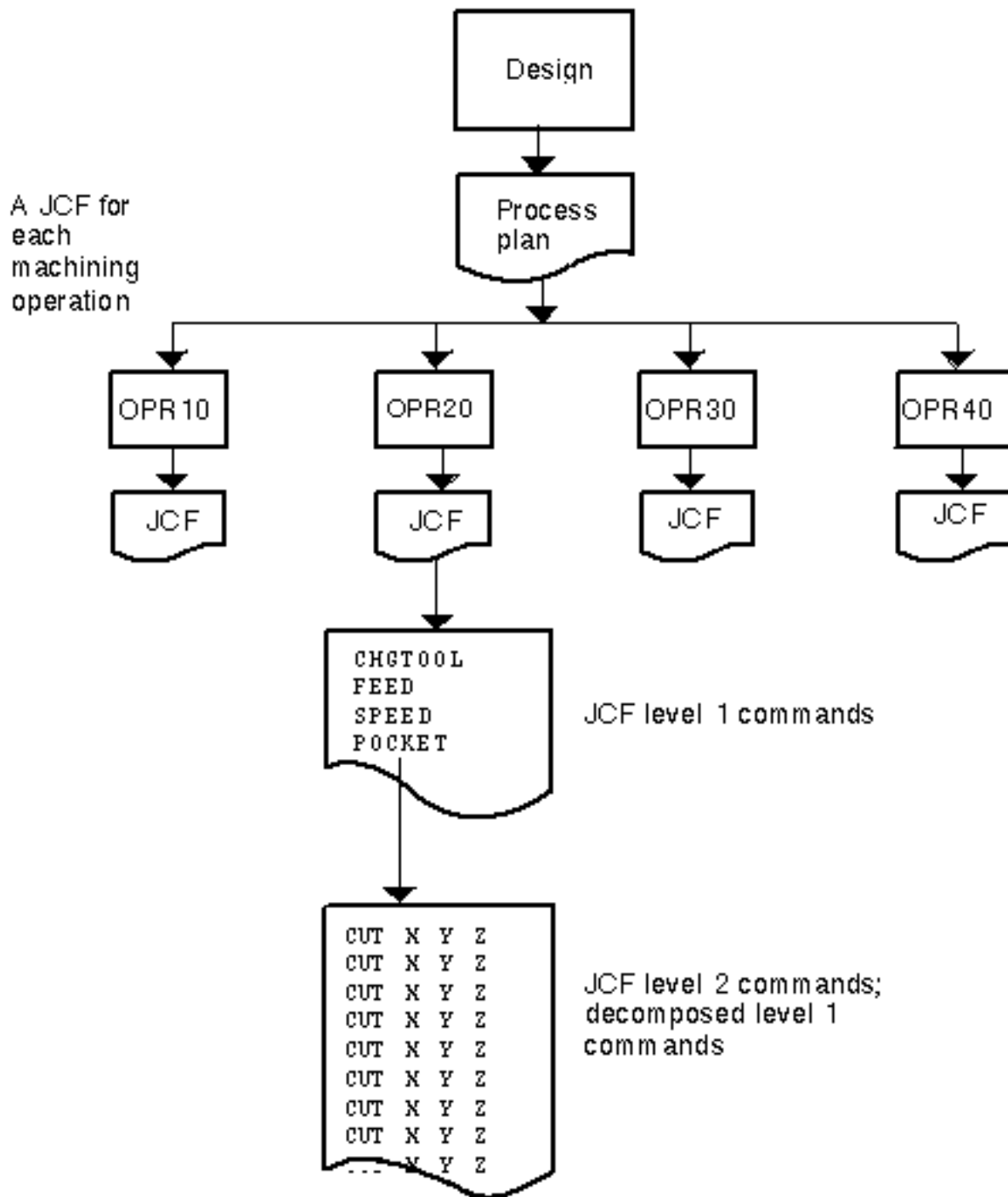


Figure 1-7 Information Hierarchy



CVNC: Integrated with CADD5

The CVNC environment runs within CADD5. The two products are integrated in the following ways:

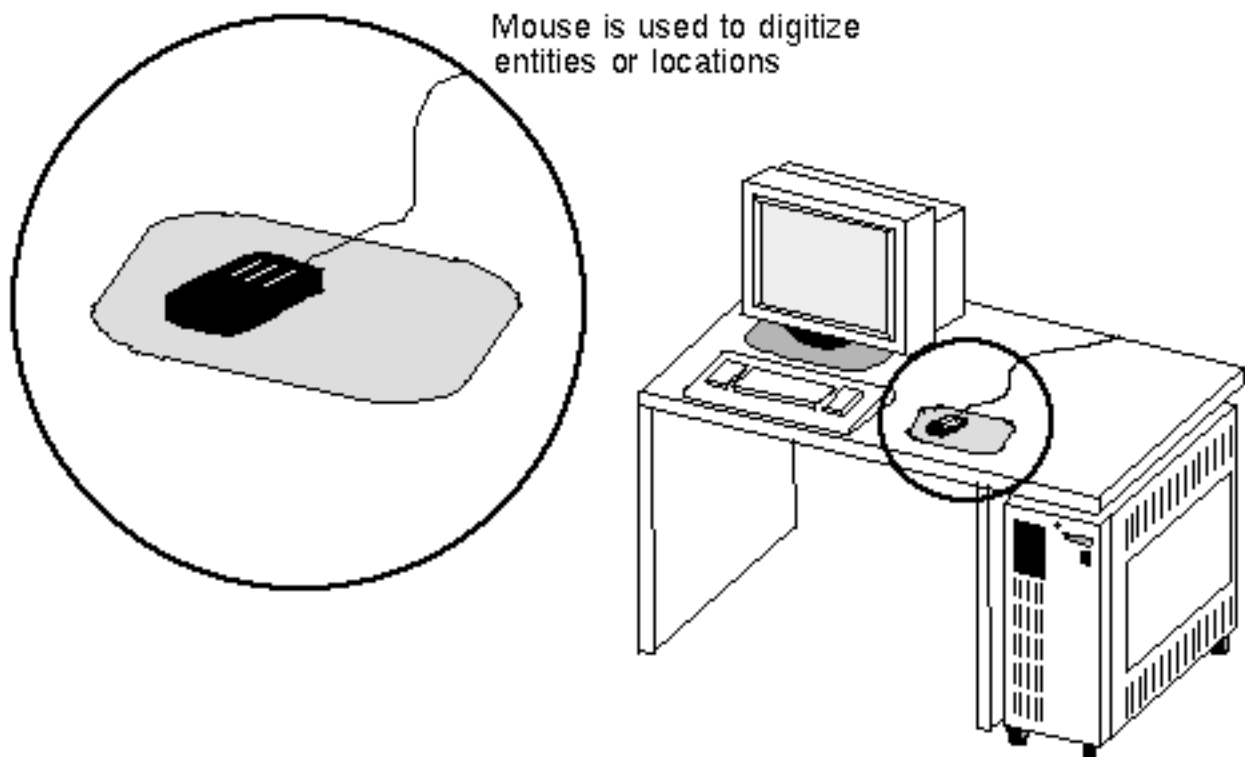
- The CVNC CADD5 interface looks and works like the CADD5 Explicit interface.
- CVNC and CADD5 share one database.
- Screen control, construction planes, and CADD5 Location and Placement options (getdata) apply in both CVNC and CADD5.
- At any time within CVNC, you can quickly access CADD5 and its functions.

Referencing CADD5 Geometry

CVNC references the active part geometry to generate tool motion. You can specify a geometric entity by either digitizing (selecting) it or tagging it (identifying with a tag) as noted in the two following figures.

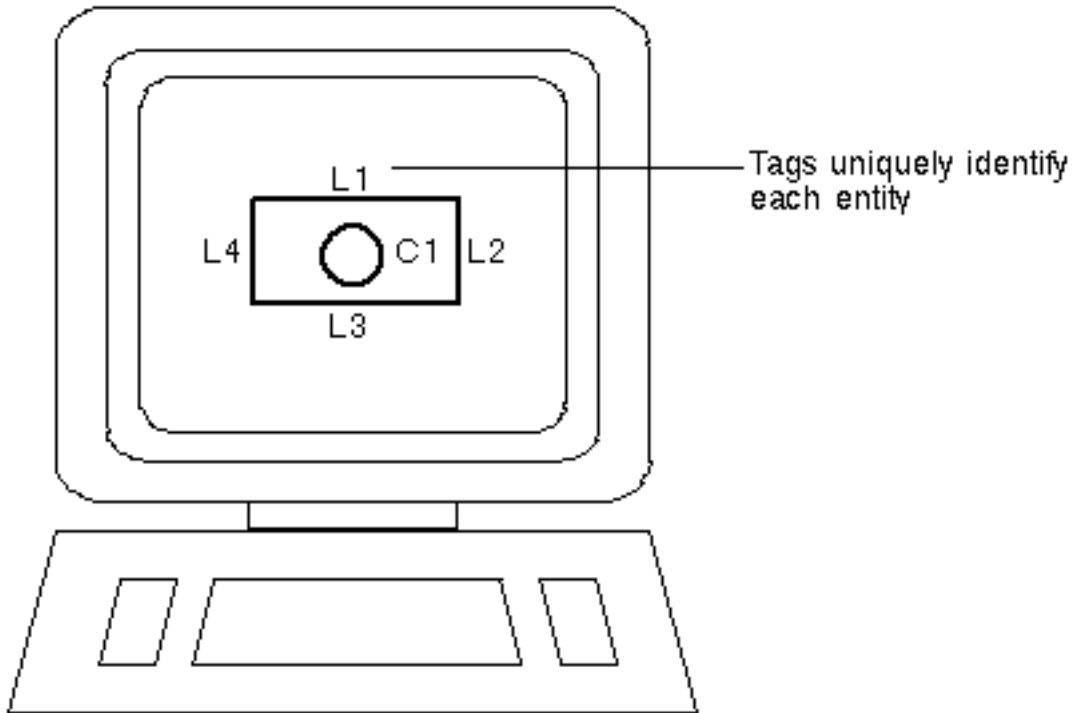
- Digitizing (selecting locations or an entity with the mouse) as shown in the following figure.

Figure 1-8 Selecting Entities



- Entering the entity tag, as shown in the following figure.

Figure 1-9 Identifying Entities with Tags



A tag is a unique entity name assigned in CADD5 or CVNC. Tags allow a CVNC part program (Job Control File) to be executed with separate versions of a part data; this minimizes additional part programming due to engineering changes. The CVNC part program records user input, CADD5 entity tags, and coordinate locations.

Changes to Geometry

Because CVNC and CADD5 (in both Explicit and Parametric environments) work with the same database, you can make changes to the part design and maintain the association between the part and its previously developed JCFs.

Tags link the geometry and CVNC. For example, if you change an entity tag, you will need to make the same changes to the corresponding JCF tag name. But, if you only change the entity size or location, the JCF can be successfully reexecuted.

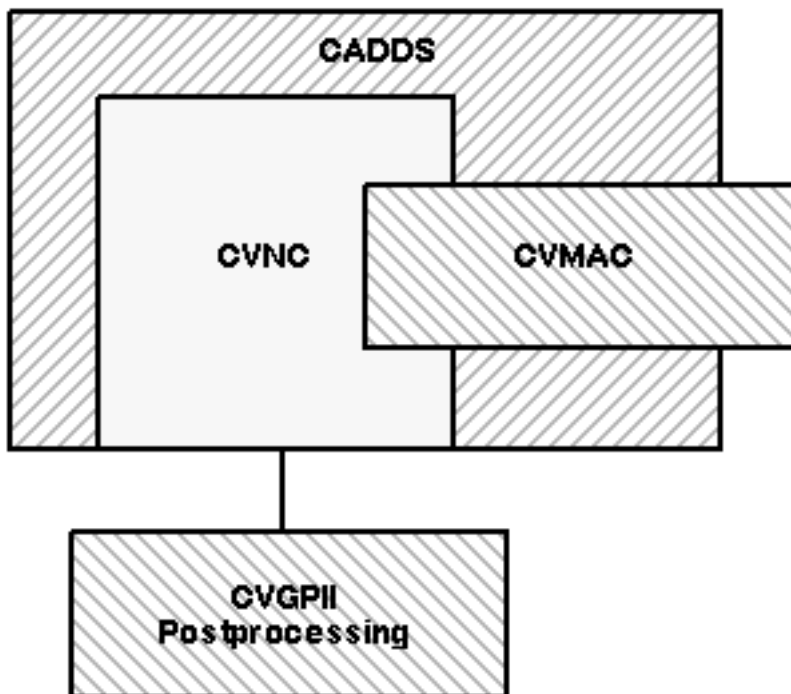
Whenever a geometry change has been made in either the Explicit or Parametric environment, you are notified in CVNC of the change and whether it necessitates reexecution to update the JCF.

CVNC: Works with a Range of Integrated Applications Software

Part programming is only one step in the numerical control machining process. CVNC is integrated with additional software to take you from design through machining and inspection:

Figure 1-10 CADDs and CVNC Integrated Software

Operating System



| | |
|---------|--|
| CADDs | CADDs allows you to develop, modify, and manipulate two-and three-dimensional model geometry. |
| CVMAC | CVMAC programming language allows you to write your own CVNC commands and install them as part of CVNC's language. |
| CVGP II | CVGP II provides a series of standard postprocessor solutions for the CVNC product family. |

We will look at each application in more detail in Chapter 3, “The Numerical Control Solution”.

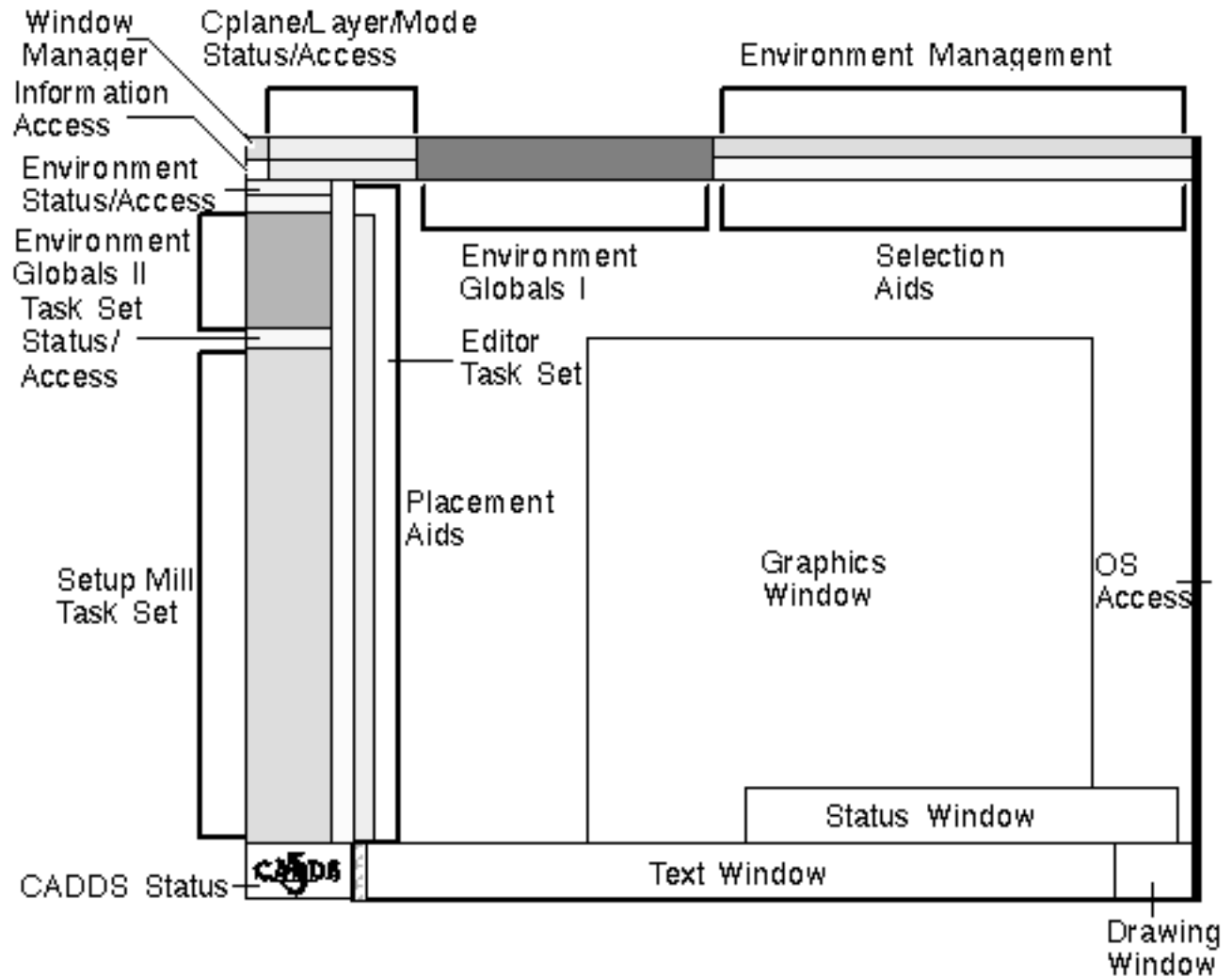
CVNC: Provides a Sophisticated User Interface

CVNC provides a state-of-the-art user interface that simplifies product use with icons, menus, property sheets, popup menus, and online documentation.

The CVNC environments desktop is divided into several distinct menu areas. Each menu area serves a specific function. In addition

- Menu selections are task oriented
- Menus are consistent across environments, where possible

The left panel icons, associated with the currently active environment, task set, and placement aids, are illustrated in detail in Figure 1-11, “Left Panel Icons,” on page 1-21.



The left panel icons are associated with the currently active environment, task set, and placement aids.

1. Environment/Group status and access identifies the current environment group, the active environment, and provides the option to resume to CADDs.
2. Environment globals II offers commonly used options: assigning status and variables, creating NCGROUPs, showing the tool, displaying status, echoing, and all tooling options.
3. Task set status and access identifies the current set of tasks, such as Setup Mill, and provides the option to select a different task set, for example, 3-Axis.
4. Task set menu includes the tasks/functions associated with a specific aspect of NC programming.
5. Entity placement aids, which include the Cartesian and Cylindrical coordinate keypad, various point-on-entity constraints, and a tag/layer toggle.
6. CVNC Editor icons, offers JCF editing options: resuming to CADDs, switching between two display modes for tag and location data or frequency of graphics updating, replacing text, redigitizing of one or a group of entities, accessing editing mode, deletion, canceling of a previous deletion, displaying of JCF lines, moving around within a JCF, and switching execution on and off.
7. Blank icon fields provide areas for you to customize with different arrangements of the icons CVNC provides or for icons you create.

CVNC: Allows User Customization

You can customize CVNC with:

- Command and execute files
- Status display
- CVMAC macros
- Pass-through statements

Command and Execute Files

Command and execute files are text files containing a command sequence. They are useful for repetitive functions, such as moving to the home point, configuring the machine, or changing the tool.

You can create a command file library and access its command files from different JCFs. For example, you may have a command file containing standard job setup commands. Rather than manually enter each command within each JCF, you can simply call the command file.

Status Display

System and machining parameters are stored in system variables. You can continuously display and monitor system variable values as they change in your JCF. The contents of these system variables appear in the Status Display window on the screen.

You can customize status displays by creating status definition files. These files tells CVNC which variables to display, as well as the format in which to display them.

CVMAC Macros

You can create CVMAC macros to be called from your JCF during an NC process. Macros can use logic, branching, and looping functions, and, therefore, provide a powerful customization tool.

To install macros into the CVNC environment, add a statement to the CVNC language identifying the macro. You can also write online documentation for macros that can be accessed in the same manner as online documentation for other CVNC commands.

Pass-Through Statements

Pass-through statements are statements passed directly to the output file; they do not affect CVNC system processing. These statements are typically used to control output processing or the machine tool. CVNC supplies a number of pass-through statements for CLFile, APT, and COMPACT II output.

You can customize CVNC by adding pass-through statements for your shop machine tools or for other specialized requirements. You can also write online documentation for your pass-through statements that are accessed in the same manner as online documentation for other CVNC commands.

CVNC Information

Accompanying your CVNC software are a number of books that describe how to use CVNC functions.

Understanding CVNC (this book)

New users or people interested in a CVNC overview read this book to acquire a general background on CVNC functions and how CVNC provides various NC solutions.

CVNC System User Guide and Menu Reference

Once you have completed *Understanding CVNC*, you are ready to begin working in CVNC. Information in the *CVNC System User Guide and Menu Reference* provides a foundation for all CVNC applications and describes specific tasks, such as building JCFs and generating output.

CVNC Applications User Guides

Along with the *CVNC System User Guide and Menu Reference*, you will need the application user guide(s) for the CVNC software you are using. For example,

If you are using CVNC-M2, use the

- *CVNC-M2 User Guide*

If you are using CVNC-M3, use the

- *CVNC-M2 User Guide*
- *CVNC-M3 User Guide*

If you are using CVNC-M5, use the

- *CVNC-M2 User Guide*
- *CVNC-M3 User Guide*
- *CVNC-M5 User Guide*

If you are using CVNC-T2, use the

- *CVNC-T2 User Guide*

If you are using CVNC-P2, use the

- *CVNC-P2 User Guide*

CVNC Editor Guide

See this book for instructions for CVNC editor commands.

CVNC Command Reference Information

Once you are familiar with CVNC, you can consult the alphabetically-arranged CVNC command references. Information includes a description for the command and its modifiers, plus a diagram illustrating how to enter command data. For example,

- If you are using CVNC-M2, CVNC-M3, or CVNC-M5, consult the *CVNC Milling Command Reference*.
- If you are using CVNC-T2, consult the *CVNC-T2 Command Reference*.
- If you are using CVNC-P2, consult the *CVNC-P2 Command Reference*.

CVNC System Variables Guide

See this book for listings of CVNC system variables.

CVNC Command and Interface Cross-Reference

See this book for a listing and brief description of each CVNC command, along with its corresponding interface access.

CVNC Work Examples

See this book for sample parts, along with annotated JCFs.

Customizing CVNC

This book explains procedures for tailoring CVNC for your site needs.

CVNC Master Index

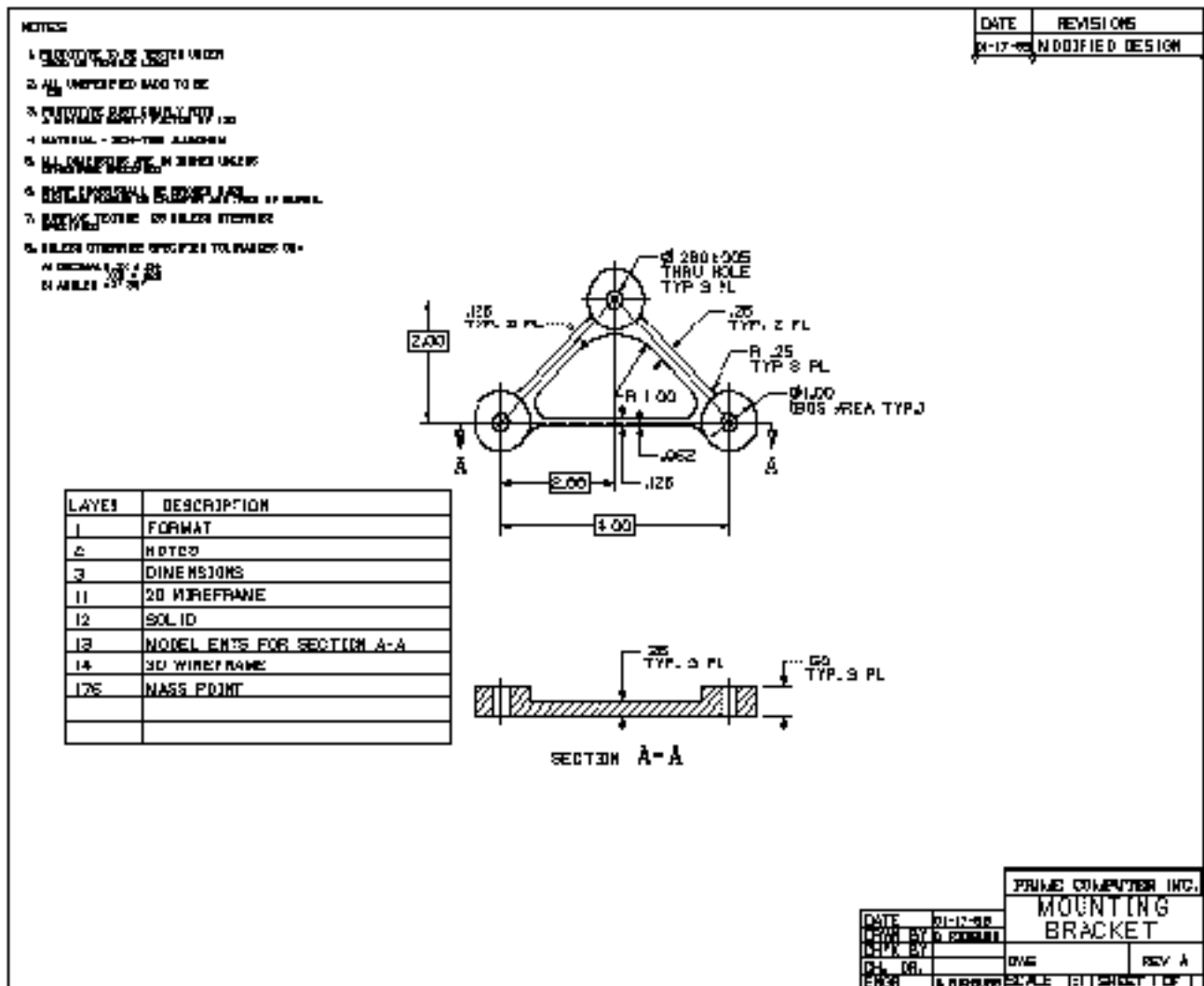
This document provides a comprehensive index for all CVNC books, in addition to other CVNC reference information.

Introducing the Sample Part

A sample part is used in Chapter 3 to illustrate the steps involved in the NC solution. This part is a mounting bracket used on the space shuttle.

Our sample part (shown below) is a mounting bracket. The bracket suspends a piece of photographic equipment in the space shuttle cargo-bay. In Chapter 3, "The Numerical Control Solution", each machining process step is illustrated, starting at design and finishing with part inspection.

Note the icon in the top right corner of this page. Each section about the mounting bracket begins with this icon.



This chapter describes the CVNC icons, menu selections, property sheets, and online documentation you can use to create and modify JCFs. A JCF is a numerical control part program that stores machine tool instructions which can be saved, edited, and reexecuted. In addition to describing general CVNC features, this chapter also outlines features that are unique to each CVNC application.

- User Interface
- CADDs Access
- Job Control File (JCF)
- System Variables
- Entity Groups
- CVNC Editor
- Tool and Tool Path Display
- Tool Path Verification
- Command Decomposition
- Command and Execute Files
- CVMAC Macros
- Output
- CVNC Application Summaries
- CVNC-M2
- CVNC-M3
- CVNC-M5
- CVNC-T2
- CVNC-P2

- CVNC Batch

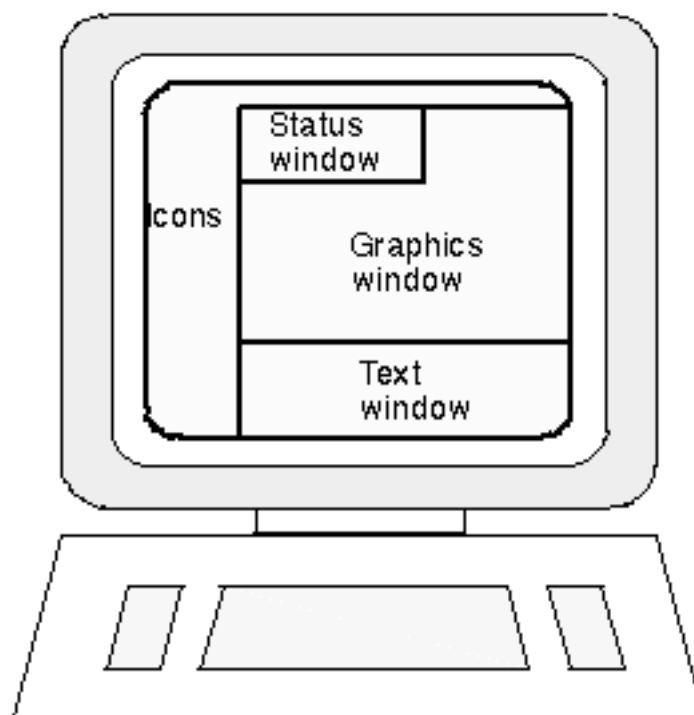
User Interface

CVNC builds on the CADD5 interface for the Explicit environment. You can use icons, menu selections, and property sheets to create and modify JCFs. Online documentation provides continuous access to information on how to use CVNC as you work.

CADD5 CVNC Interface

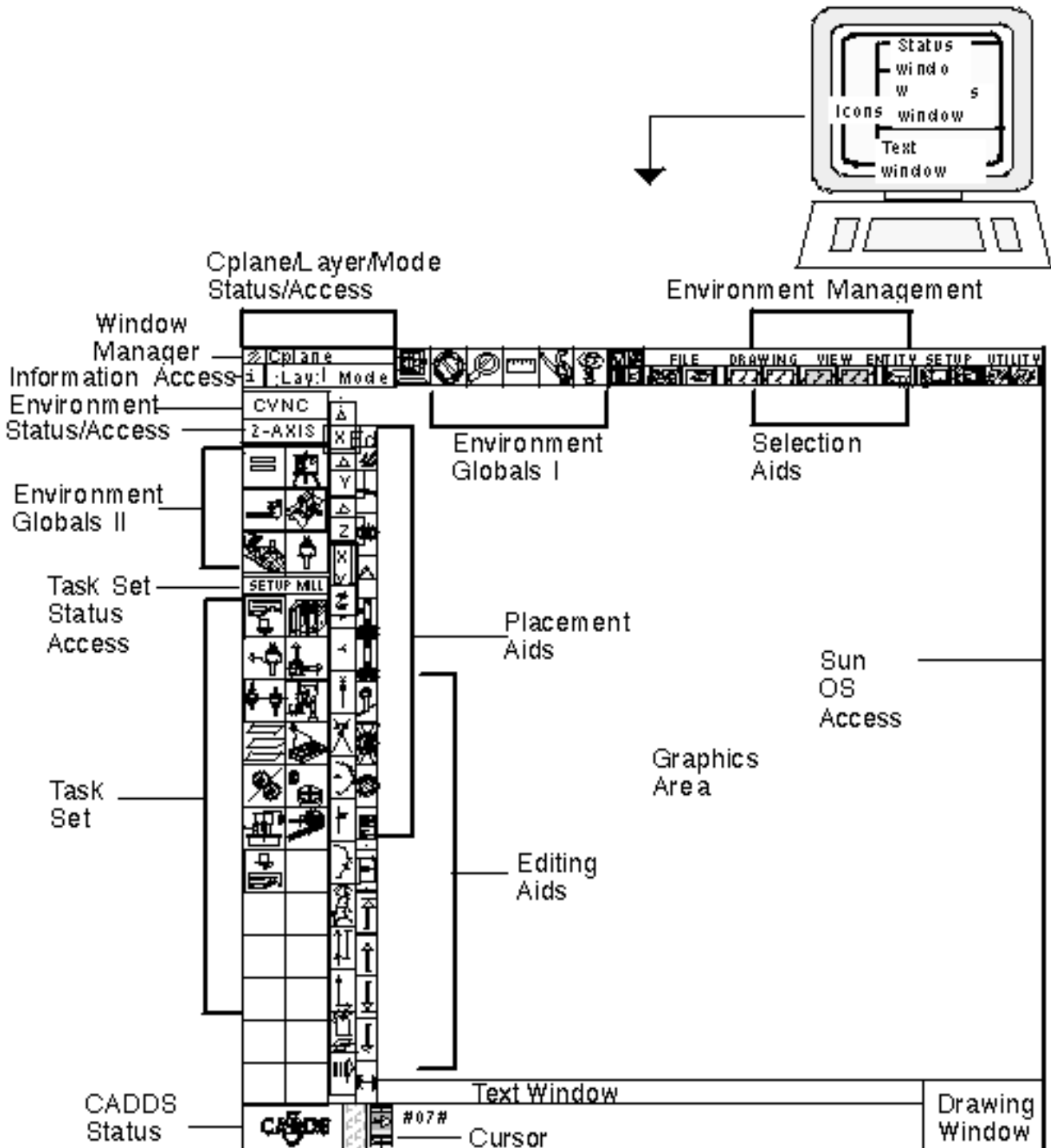
In addition to its own icons, menus, and property sheets, CVNC shares the CADD5 Explicit interface, which consists of multiple windows, as shown below. These windows allow you to interact with the system in several functional areas simultaneously. You can resize and move windows as needed. While in CVNC, you still have access to CADD5. Location and placement options (getdata) are available.

Figure 2-1 Multiple Windows



The illustration Figure 2-2, “Desktop Layout,” on page 2-4 shows the general layout of the interface. While in CVNC, you have easy access to your part, CADD5, online information, and system functions, in addition to the CVNC application you are working on (in this case, CVNC-M2). For more information, see the *CVNC System User Guide and Menu Reference*.

Figure 2-2 Desktop Layout



Icons and Menus

Icons, menus, and property sheets allow you to quickly and easily access CVNC and CVNC commands.

The CVNC interface, like CADD5, provides a set of icons in a top-down structure.

After you select the MANUFACTURE option, a property sheet is displayed that allows you to easily enter a new or previously-existing JCF.

Previously-existing JCFs are listed on your screen; you can simply highlight your selection and proceed.

The Figure 2-3, “Sample CVNC Icon Task Set: Editing JCFs,” on page 2-7 illustrates an icon task set for CVNC-M5, along with the icon to establish z-planes.

Each icon accesses a property sheet that guides your input into the JCF. In addition, you can use a pulldown menu which automatically inserts the initial key words of a CVNC command.

Command input is then inserted in the CADD5 text window and automatically stored in the JCF.

The CVNC interface allows you to

- Access CVNC applications loaded on your system
- List, edit, and delete a JCF
- Execute CVNC commands
- Return to CADD5

Note the empty icons in the bottom right corner. These are provided for you to customize the interface with additional or reorganized task icons to suit your needs.

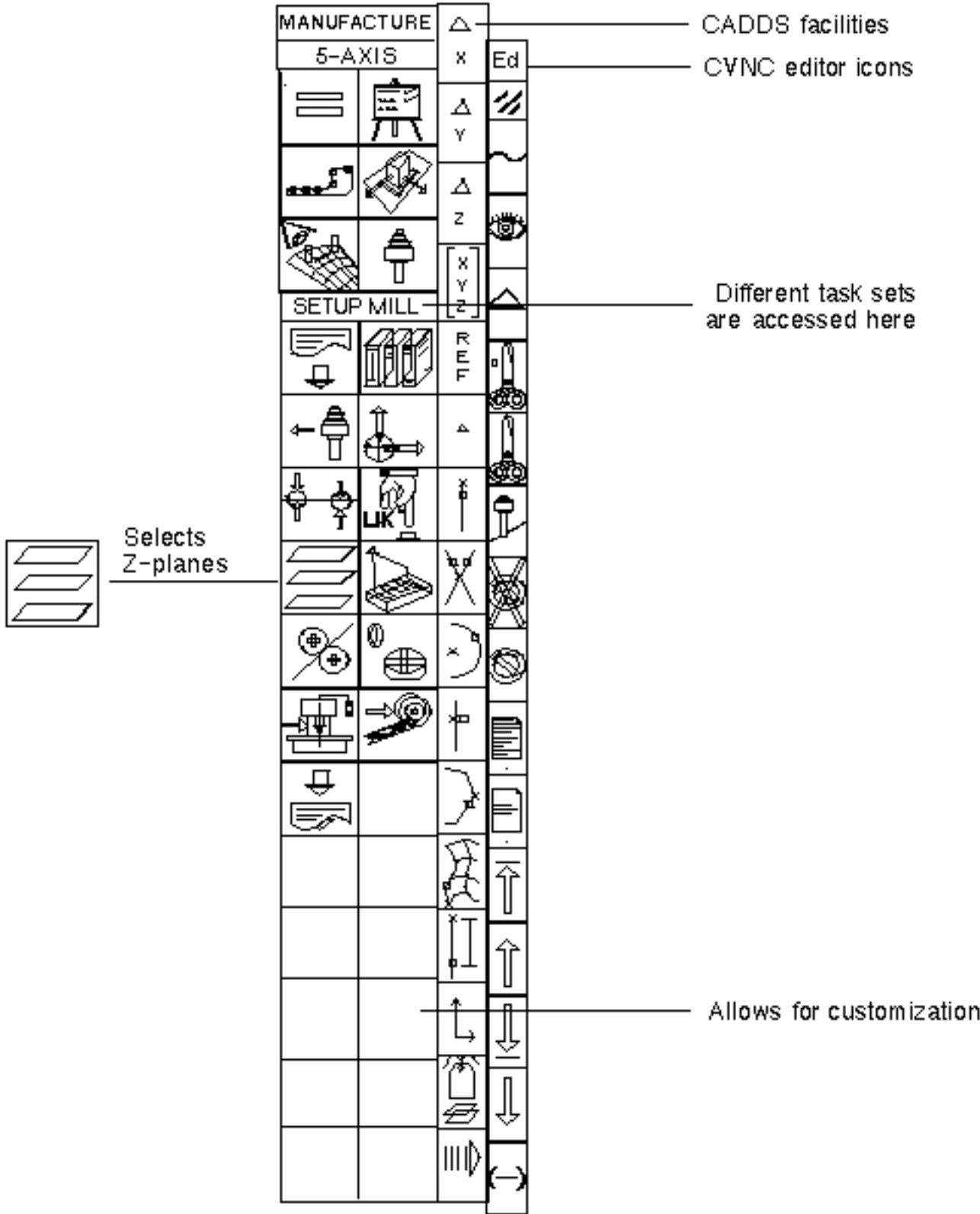
Editing Facilities

A strip of icons that allow you to easily edit your JCF lines is always available. These icons are illustrated in Figure 2-3, “Sample CVNC Icon Task Set: Editing JCFs,” on page 2-7.

For example, you can

- Add information to existing lines
- Delete information from existing lines
- Step back or forward in the JCF

Figure 2-3 Sample CVNC Icon Task Set: Editing JCFs



Property Sheets

Once you have selected the Define Planes icon, a property sheet appears which allows you to set up the locations of planes the part program needs.

Highlight your selection and/or enter information requested, select the Apply option, and the CVNC command is automatically written in the CADDSS text window and executed.

Figure 2-4 CVNC Planes Property Sheet

The image shows a dialog box titled "Define Planes". It contains several sections for configuring plane definitions:

- Plane to define:**
 - Clearance plane
 - Retract plane
 - Work plane
 - Approach plane
 - Reference plane
- Define plane location:**
 - Define plane location
 - z
 - Absolute
- Relative to:**
 - Select location
 - Clearance plane
 - Retract plane
 - Work plane
 - Approach plane
 - Reference plane
- Offset from current location:**
 - Offset from current location
 - dz
 - Up Down

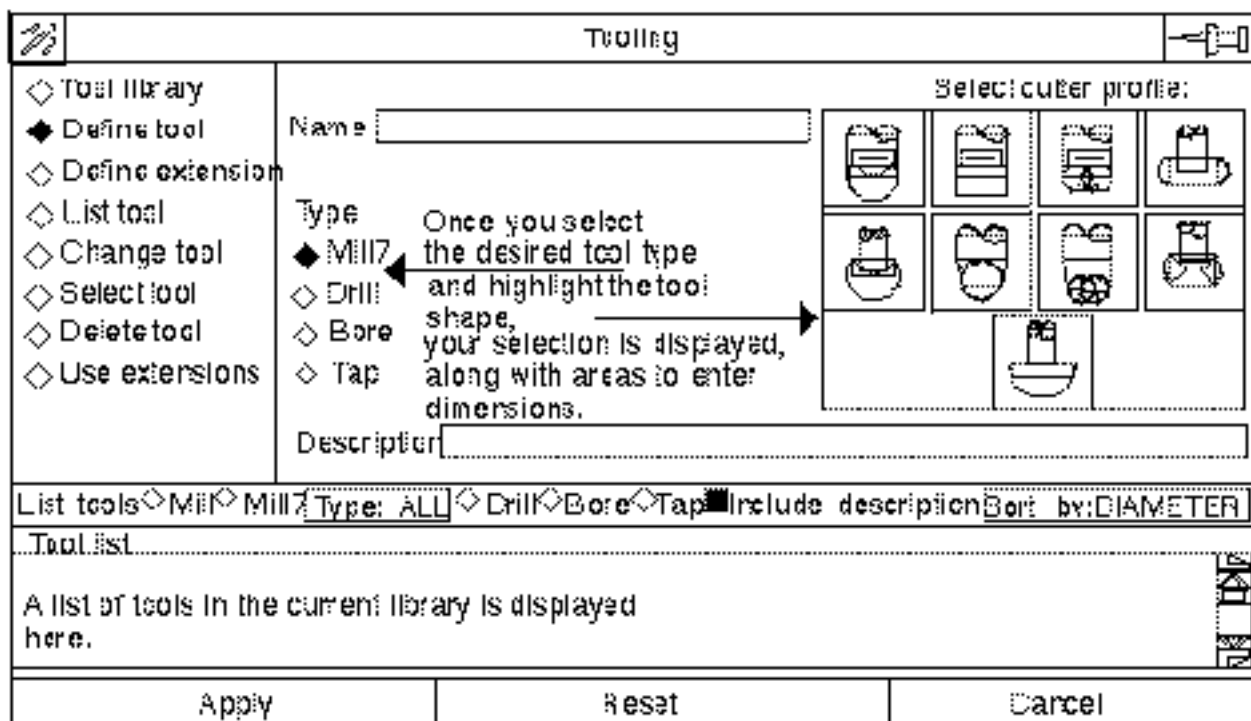
At the bottom of the dialog are three buttons: "Apply", "Reset", and "Cancel".

Tooling Management

The tool icon accesses two property sheets that provide an array of options. The property sheet illustrated below allows you to

- Set up tool libraries to store tool definitions. In subsequent JCFs, you do not need to redefine tools; you need only activate the appropriate tool library.
- Define tools and tool extensions, and view the tool shapes and dimensions your selections create.
- List tools, and then highlight selections to be activated or removed.
- Automate tool and tool extension selection within the JCF.

Figure 2-5 Tooling Property Sheet



Online Documentation

Online documentation is easily accessible from within CVNC. It provides descriptions of CVNC commands, command modifiers, editor commands, and system variables. For more information, see the *CVNC System User Guide and Menu Reference*.

You can access online documentation to

- Open up a documentation window providing a description of the command and a sequential list of its modifier descriptions (see Figure 2-6, “Online Documentation Window,” on page 2-11).
- List all CVNC commands in alphabetical groupings in a popup menu in the graphics window and in the status window (see “List of Commands” on page 2-11). You can scroll through menus of all the commands.
- List all commands starting with a specific letter in a popup menu and in the status window (see “List of Commands Starting with ‘S’” on page 2-11).
- List all modifiers available to a command. This list appears in a popup menu in the graphics window and also in the status window.

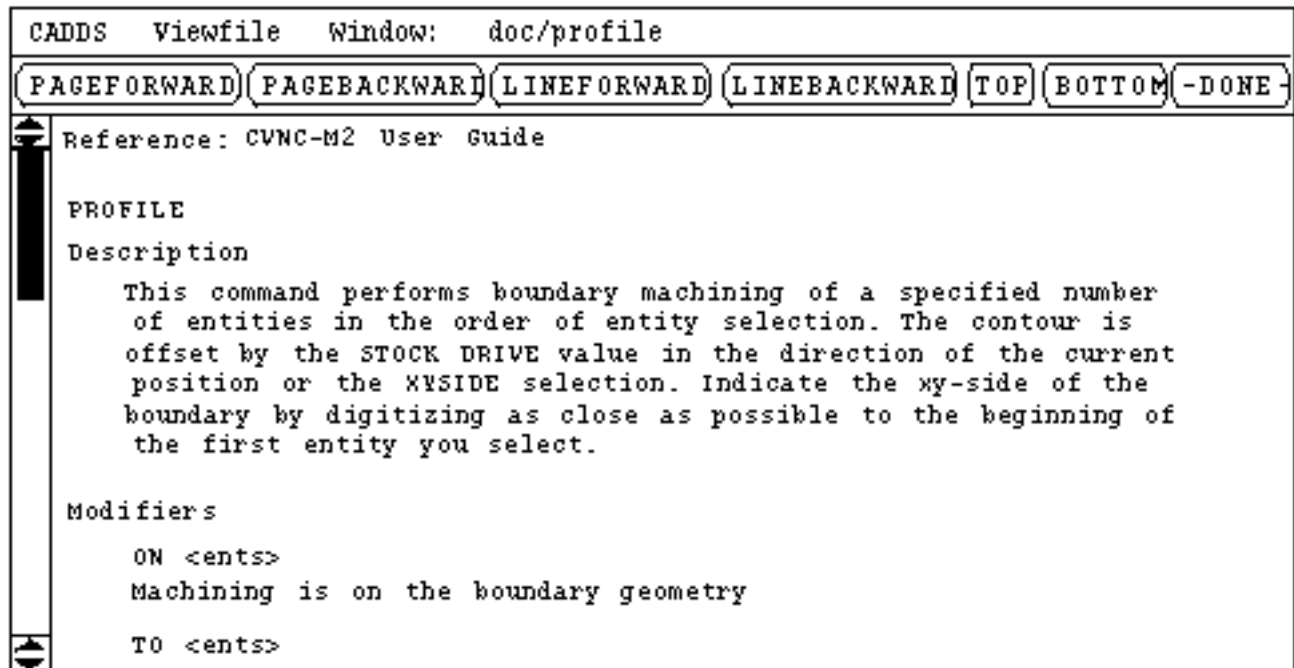
Documentation Window

When you access documentation for a command, a window automatically opens (see Figure 2-6, “Online Documentation Window,” on page 2-11). This window displays

- Command and modifier descriptions
- References to appropriate hardcopy documentation
- Buttons that scroll text forward, scroll text back, and close the window to quit online documentation

Please note: You can also read online documentation in its text file format from an OS shell. These online documentation files can be customized to include more information on a command, such as how it relates to your shop’s procedures.

Figure 2-6 Online Documentation Window



List of Commands

```
CADDSPATH          Part format is: SINGLE
CADDSPATH/PART:   SAMPLEDRAW: 1
CHANGED: 7-18-91  Model Unit: IN Draw Unit: IN
Mode: Model       Input Unit: IN
CPLANE: TOP LAY:0 Environment: General CADDSPATH
NC options : ACCELAPPROACHAREAMILLASSIGN  AUDIT
NC options : AUXFUNBLOCKBOREBREAK  BRING
NC options : CALCRADCALLCHECKLIBCHGTOOL CLEAR
```

List of Commands Starting with 'S'

```
CADDSPATH          Part format is: SINGLE
CADDSPATH/PART:   SAMPLEDRAW: 1
CHANGED: 7-18-91  Model Unit: IN Draw Unit: IN
Mode: Model       Input Unit: IN
CPLANE: TOP LAY: 0 Environment: General CADDSPATH
NC options : SELCTLSELECTSEQNOSET  SETAX
NC options : SHOWSPEEDSPINDLSPROF  STATUS
NC options : STOCKSTOP
```

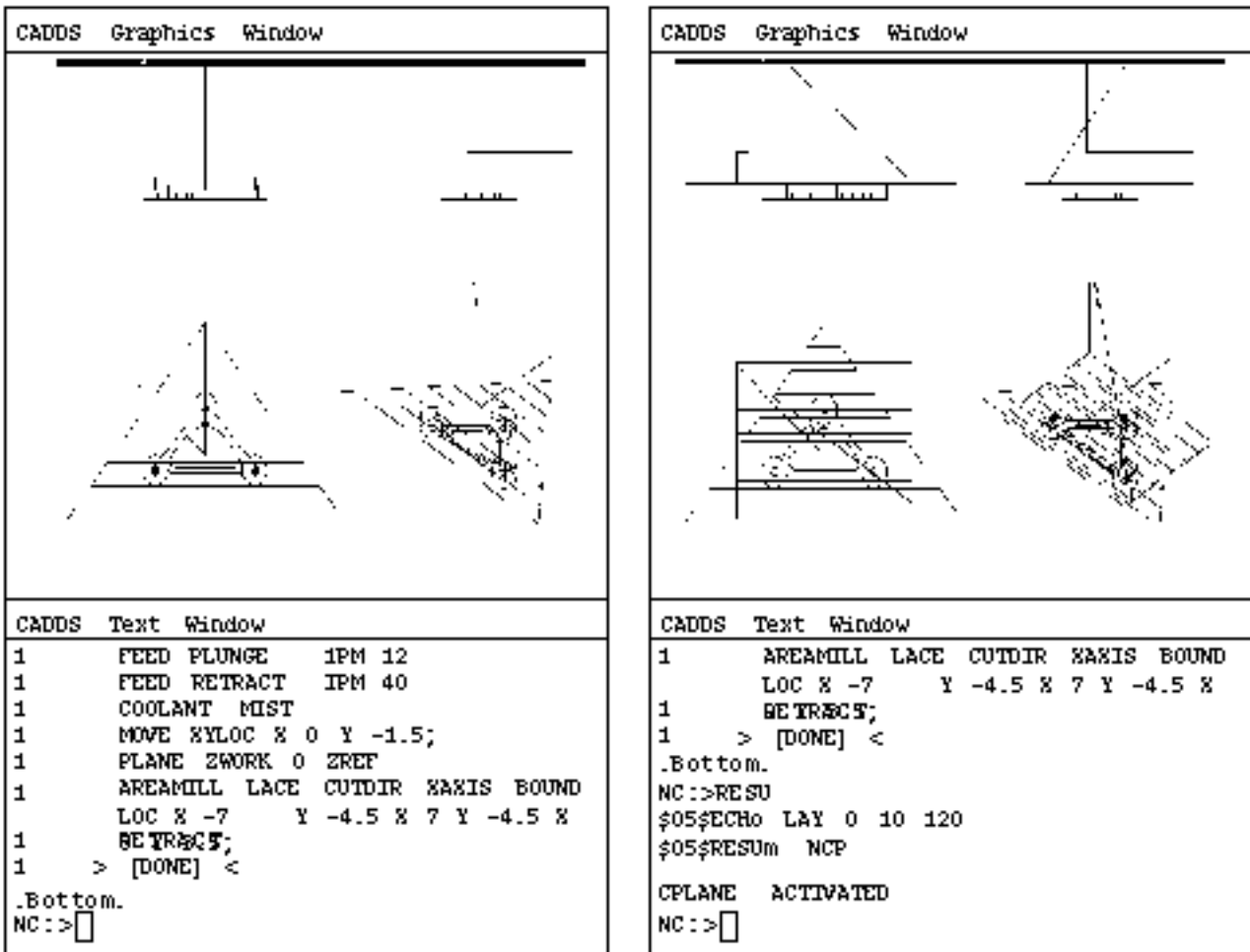
CADDs Access

CVNC allows easy access to CADDs functions. You can access CADDs from anywhere within a JCF using the Tilde function (~) or the CVNC RESU or IC (Insert CADDs command) commands. These commands place you back at the CADDs prompt. You can execute CADDs commands and then return to CVNC.

For more information, see the *CVNC Editor Guide*.

In the figure below, another layer is echoed while accessing CADDs with RESU.

Figure 2-7 Fast CADDs Access



Job Control File (JCF)

A JCF is a numerical control part program that stores a sequence of instructions to control the machine tool. These instructions include

- CVNC commands
- Macros
- Pass-through statements

A JCF is an ASCII text file that can be saved, edited, and reexecuted. The JCF is a permanent NC part program.

As you enter CVNC commands, the JCF is automatically generated, providing graphics as appropriate. As you execute each command, you can examine and verify the resulting tool motion. Typically, a JCF corresponds to an operation or a single setup on a machine tool. The JCF consists of commands that result in output for machine control (such as CHGTOOL and FEED), and tool motion (such as CUT, MOVE, and POCKET).

A full set of editing commands is available to position you within the JCF and make changes, such as changing, copying, and deleting lines. These edits are simultaneously reflected in the graphics display.

The display on the opposite page shows a sample JCF. Chapter 3, “The Numerical Control Solution” shows how this JCF was created.

How Are Job Control Files Created

You can create a JCF in three ways:

- **Interactive** - Commands are immediately verified for correct syntax, executed, and stored in the JCF. Motion commands result in graphic display on your screen.
- **Syntax Verification** - Commands are checked for correct syntax and stored in the JCF, but they are not executed immediately. JCF command execution can be performed at a later time, perhaps during off-hours.
- **Offline** - Commands are written into a file using a text editor. They can subsequently be executed interactively or in batch mode. There is no automatic syntax checking performed when this method is used. There is syntax checking when these commands are executed.

Figure 2-8 Job Control File Sample

```
[JCF  CVNC MILL  ]
AUDIT ON
#>*
#>*  SIDE 1 - OPERATION 1
#>*
#>*  MILL STOCK FLAT, APPROXIMATELY A .010" CUT.
#>*
#>*  WHEN THIS OPERATION IS COMPLETE TURN PART OVER.
#>*
OUTPUT CLFILE TEXT 'BOTTOM-MILL'
LAY 105
PLANE ZREF 0
PLANE ZAPPR 1.04 ZREF
PLANE ZCLEAR 1 ZREF
PLANE ZRETRACT .05 ZREF
PLANE ZWORK -.5 ZREF
DEFTOOL 'MILL2-1' MILL7 DIA 1 CORNER 0 HEIGHT 5
HOMEPT FROM X -3 Y 1 Z 5;
CLEAR HOME
CHGTOOL 3 'MILL2-1'
SPEED RPM 1800 CLW
FEED APPROACH IPM 80
CALL PLANES
FEED CLEAR RAPID
FEED CONNECT IPM 20
FEED CUT IPM 20
FEED PLUNGE IPM 12
FEED RETRACT IPM 40
COOLANT MIST
PLANE ZWORK 0 ZREF
AREAMILL LACE CUTDIR XAXIS BOUND LOC X -4 Y -4 X 4 Y -4 X 4 Y 2 X -4
Y<*
 2;
RETRACT
COOLANT OFF
CLEAR HOME
[DONE]
```

For more information, see the *CVNC System User Guide and Menu Reference*.

System Variables

System and machining parameters are stored in system variables. The status of these variables can be displayed in the CADD5 text or status windows. You can also use system variables to pass information to CVMAC macros.

CVNC System Variables

CVNC has a large number of system variables that contain the current state of machining and CVNC system parameters. These contain data, such as feed rate, spindle speed, and tool radius.

There are three types of system variables: numeric, text, and coordinate. A numeric variable represents one value (such as feed rate), a text variable represents a text string (such as a file name), and a coordinate variable represents three values (the x-, y-, and z-axes). The pound symbol (#) is the first character of the names used to reference system variables.

Displaying Variables Status

You can display the current values of system variables in the Text Window using the SHOW or PRINT command.

```
;
1      RETRACT
1      > [DONE] <
      Bottom ----- Shows the current location of the tool
NC:>SHOW #CURLOC;
CO-ORDINATES:      -5.898116      -4      0.05
NC:>PRINT #FEDRAT— Prints the current feed rate
40
NC:>
```

System variables can also be displayed in the status window. You can define the format of the status display with a status definition file. You can also turn different status displays on or off within a session.

```
STATUS BLOCK NAME : A
Tool : Loc X -3      Y 1      Z 5
Speed: Value 3600Units RPMDir CLW
Feed : Value 0Units RAPID Mode CLEAR
Plane: Zwrk -0.5 Zclr 1 Zapp 1.04 Zret 0.05
```

For more information, see the *CVNC System User Guide and Menu Reference*.

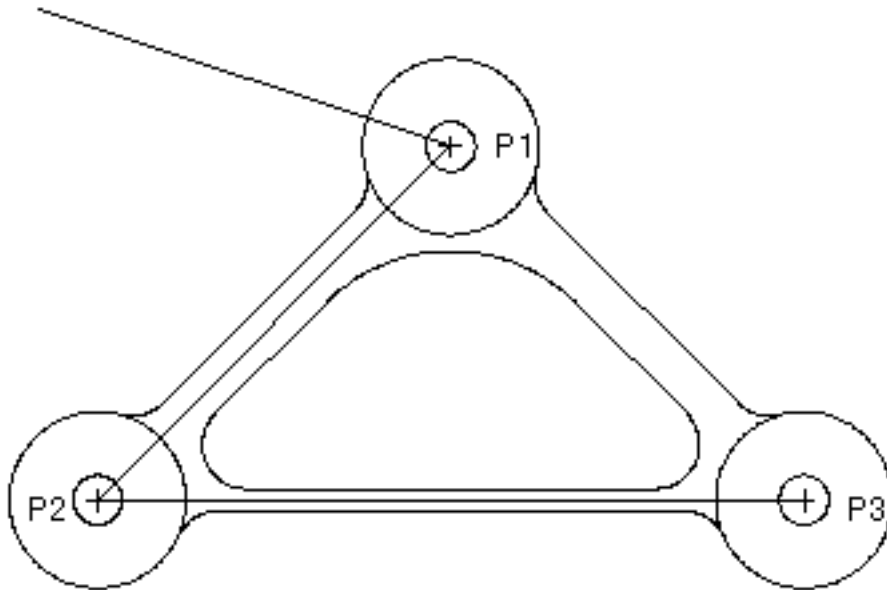
Entity Groups

You can use the same entities or locations in multiple operations, such as pocketing/profiling or drilling/countersinking/boring/tapping. You can do this more efficiently by grouping entities together and naming them. For example, you can group three holes that are to be drilled and tapped. You can use tags or digitizes to define an entity group (up to 512 entities or locations can be grouped).

```
NC:> NCGROUP HOLES ENT $P1 $P2 $P3
```

Throughout the JCF, instead of entering digitizes or tags to reference those three holes, simply enter the NCGROUP name.

```
NC:> DRILL DEPTH .5 ENT HOLES;
```



Within CVNC-M3 and CVNC-M5, you can group surfaces and wireframe entities into surface lists. This is done using the NCGROUP command with the SURF modifier. Attributes (such as machining stock) can be assigned to surface lists.

Once an NCGROUP is created, you can highlight its contents during your CVNC session with color or font change.

For more information, see the *CVNC System User Guide and Menu Reference* and the *CVNC-M3 User Guide*.

CVNC Editor

The CVNC editor allows you to move within the JCF and modify CVNC commands and their sequence. Command input is the default mode in CVNC; therefore, you are not normally using the editor when entering CVNC commands. However, the editor provides commands to control and modify your JCF, giving you greater control of the part program.

Editor commands do not become a permanent part of your JCF, they are only used to change or manipulate it. For example, you can step backward and forward in the JCF to modify, insert, or delete commands.

When you step back and forth in your JCF, commands are only reexecuted if data has changed, making positioning quicker and more efficient.

Editor Facilities

A range of advanced editing facilities are built on the two basic actions, Forward and Backward, that allow you to manipulate individual JCF lines and groups of JCF lines. “Editor Commands” on page 2-17 illustrates a few editing facilities.

Updating Graphics

CVNC updates graphics as you position in the JCF. For example, you can move backward in your JCF, watch the associated tool path and tool graphics disappear, make changes with the editing commands, and move forward over the changed commands. CVNC maintains the current tool path graphics accordingly.

CVNC Star Buffers

The editor also has an advanced facility, called star buffers, that allows you to store groups of editor commands in sequence and then repeatedly call and execute them during your CVNC session.

Editor Commands

```
1      FEED PLUNGE      IPM 12
1      FEED RETRACT    IPM 40
1      COOLANT MIST
1      PLANE ZWORK 0   ZREF
1      AREAMILL LACE CUTDIR XAXIS BOUND LOC X -4 Y -4 X 4 Y -4
      X 4 Y 2 X -4 Y
continue 2;
```

```

1      RETRACT
1      COOLANT OFF
1      CLEAR HOME
1      [DONE]
      Bottom
NC:>B/HOMEPT ----- Move backward to the
1      ^> HOMEPT FROM X -3 Y 1 Z 5;      HOMEPT command.
      <
NC:>E ----- Edit the command line
      HOMEPT FROM X -3 Y 1 5;      to change the value
              7      of the Z-axis.
      HOMEPT FROM X -3 Y 1 Z 7;

NC:>F/AREAM ----- Move forward to the AREAMILL command.
1      CLEAR HOME
1      CHGTOOL 3 'MILL2-1'
1      SPEED RPM 1800 CLW
1      FEED APPROACH IPM 80
1      FEED CLEAR RAPID
1      FEED CONNECT IPM 20
1      FEED CUT IPM 20
1      FEED PLUNGE IPM 12
1      FEED RETRACT IPM 40
1      COOLANT MIST
1      PLANE ZWORK 0 ZREF
1      > AREAMILL LACE CUTDIR XAXIS BOUND LOC X -4 Y -4 X 4 Y -4
X 4 Y 2 X -4 Y
continue2;
<
NC:>P15 ----- Page 15 lines of the JCF.
1      FEED CLEAR RAPID
1^-----^FEED CONNECT IPM 20
1      FEED CUT IPM 20
1      FEED PLUNGE IPM 12
1      FEED RETRACTIPM 40
1      COOLANT MIST
1      PLANE ZWORK 0 ZREF
      > AREAMILL LACE CUTDIR XAXIS BOUND LOC X -4 Y -4 X 4 Y -4
X 4 Y 2 X -4 Y
continue2;
1      RETRACT
1      COOLANT OFF
1      CLEAR HOME
1      [DONE]
      Bottom
NC:>

```

For more information, see the *CVNC Editor Guide*.

Tool and Tool Path Display

You can display and control tool and tool path graphics.

Displaying Tools

You can display tool motion. Modifier options allow

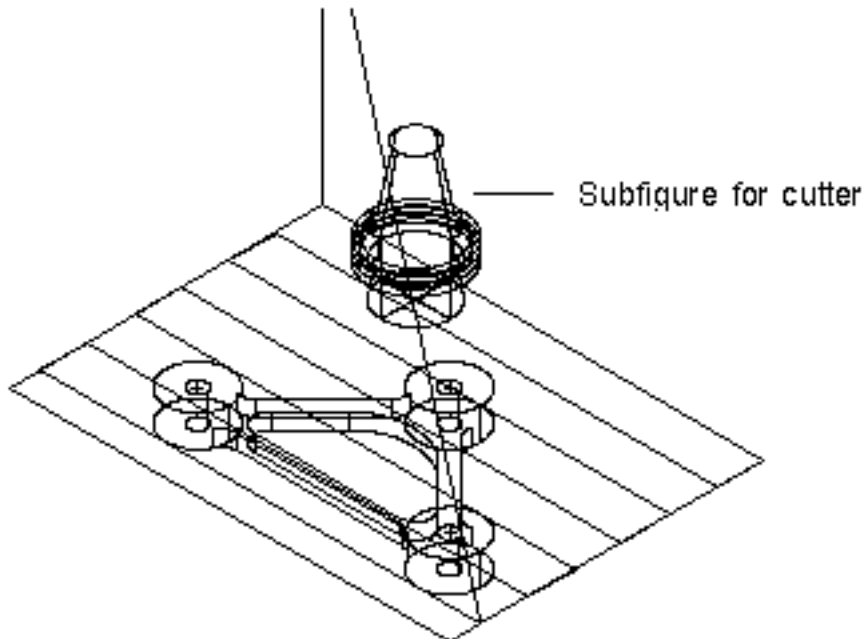
- Dynamic tool display at specific intervals.
- Static tool display at specific intervals. (You can retain previously displayed tools as motion continues.)

In addition, you can

- Turn the display on or off.
- Display a default tool or create customized tool displays.
- Erase displayed tools.
- Display the tool at vertices or at specific points along a tool path

The following figure displays a subfigure for a mill.

Figure 2-9 Tool Display with Subfigure



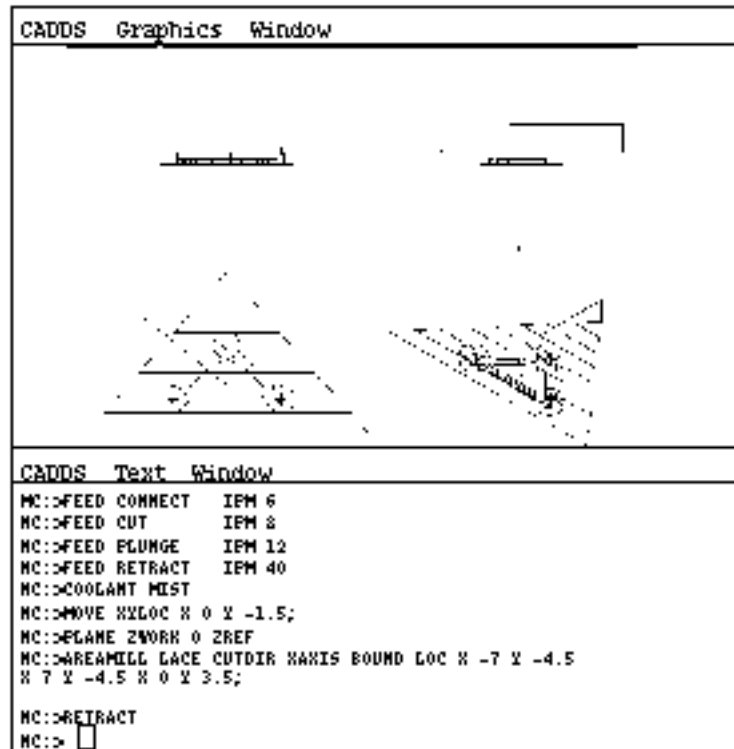
Displaying Tool Paths

The tool path is automatically displayed as you move through the JCF. The tool path is updated to correspond to the new JCF location.

The following figure shows how a lace pocketing (AREAMILL) tool path disappears when the editor command B (Backward) positions the JCF back two lines. For more information, see the *CVNC System User Guide and Menu Reference*.

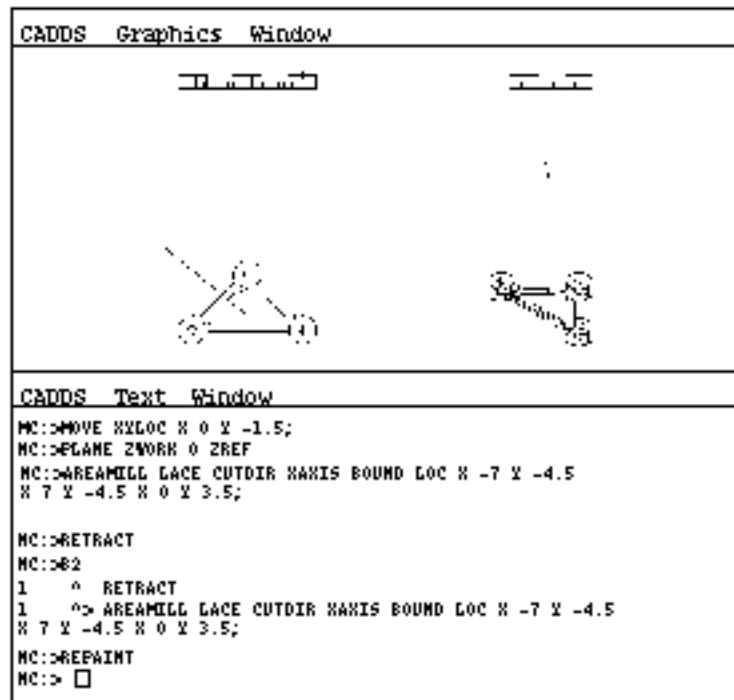
Figure 2-10 Tool Path Display

AREAMILL tool path is displayed.



B2 causes the JCF to position back two lines, erasing the tool paths for AREAMILL and RETRACT.

The REPAINT command redisplay system graphics.



Tool Path Verification

The NC Tool Path Verifier uses stock and tool path descriptions to create a realistic image of a machined part by visually simulating metal removal. This allows you to verify tool paths before submitting the NC program to a machine tool.

NC Tool Path Verifier Features

The NC Tool Path Verifier graphically simulates the milling process by displaying a wireframe or full depth image model of a work piece as it is being machined.

The workpiece is shown in blue, the tool image is displayed in green, and the background color (either gray or red) is dynamically changed in the area of material removed.

In this way, you can visually inspect the tool path progress for potential problems without creating prove-out tapes.

Please note: The NC Tool Path Verifier is a separate option available for CVNC-M2, CVNC-M3, and CVNC-M5 (includes 4-axis capability).

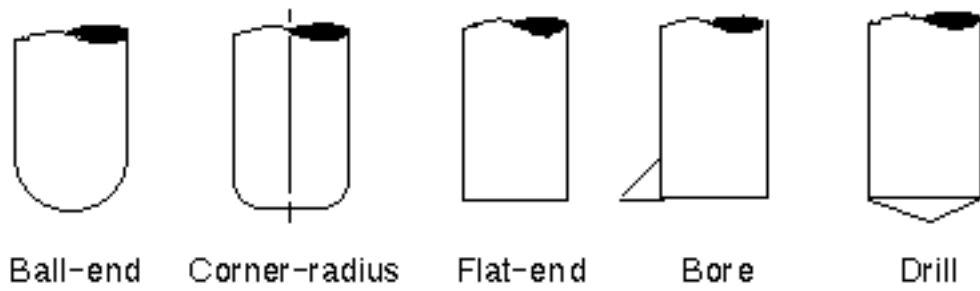
Following an initial verification to check for collisions, gouging, or nonmachined areas, you can compare the tool path results with the original geometric model within a specified tolerance.

A sectioning feature allows you to view a cross-section of the part to check for potential problems within deep cavities or behind obstructions.

Several tool paths can be verified at a time and the intermediate display saved at any point. These file images can later be used as a start point for further verification or later reference.

A sample `ncverify` window display is illustrated in Figure 2-11, “Sample NC Tool Path Verifier Display,” on page 2-23.

Figure 2-11 Sample NC Tool Path Verifier Display



For more information, see the *CVNC System User Guide and Menu Reference*.

Command Decomposition

By using CVNC commands to decompose command files, macros, and high-level commands, you can examine:

- Individual commands within command files
- Commands that result from macros
- Motion resulting from a high-level command

For example, you can decompose a turning operation, analyze the individual cuts, and make changes to the program. The following TURN command decomposes into FEED, CUT, MOVE, and APPROACH statements. Note the numbers in the left margin of the JCF. The decomposed lines are marked with 2. Depending on the command, you may be able to step in another level and further decompose a level 2 command to level 3.

```
1      [TURN $L3 $L4 $L5 $L6 $L7 $L8 $L9 $L10; MATL $L2 $L1;]
continue [MAXDEPTH .8 MINDEPTH .3 SAFDIST .1]
2      > [#>] <
2      APPROACH XABS 9.0000 YABS 7.0000 ZABS 0.0000
2      APPROACH XABS 9.0000 YABS 5.7000 ZABS 0.0000
2      APPROACH XABS 9.1000 YABS 5.7000 ZABS 0.0000
2      FEED IPM 80.0000
2      CUT XABS 0.5000 YABS 5.7000 ZABS 0.0000
2      CUT XABS 0.5000 YABS 6.0000 ZABS 0.0000
2      CUT XABS 0.5000 YABS 6.5000 ZABS 0.0000
2      .
2      .
2      FEED IPM 60.0000 )
2      CUT XABS 9.1000 YABS 2.2000 ZABS 0.0000
2      MOVE XABS 9.1000 YABS 0.5000 ZABS 0.0000
2      FEED IPM 80.0000
2      CUT XABS 6.5000 YABS 0.5000 ZABS 0.0000
2      CUT XABS 6.5000 YABS 1.3000 ZABS 0.0000
2      FEED IPM 20.0000
2      CUT XABS 6.5000 YABS 1.4000 ZABS 0.0000
2      [EOF] <
```

For more information, see the *CVNC Editor Guide*.

Command and Execute Files

Command and execute files store command sequences that are used repetitively. These files can be referenced from subsequent JCFs requiring that same command sequence.

Command and execute files store commands in a file that can be executed at a later time.

For example, this command file changes tools after prompting for the tool name.

```
[CMFIL GETTOOL ]  
READ "TOOL NAME?" TOOLNAM  
READ "TOOL NUMBER?" TOOLNUM  
CHGTOOL TOOLNUM TOOLNAM  
[EOF]
```

When you call the GETTOOL command file in CVNC, the GETTOOL command file runs as follows:

```
NC:> CALL GETTOOL  
TOOL NAME? "1/4-MILL"  
TOOL NUMBER? 2  
NC:>
```

For more information, see the *CVNC Editor Guide*.

CVMAC Macros

Macros are procedures written in the high-level language CVMAC. Macros can contain logic, branching, and looping functions to build additional intelligence into CVNC. For example, a macro may perform a set of moves at a specified depth, and then step down several times, repeating these moves until the required depth is reached. CVNC supports four macro types:

- Command macros are called from CVNC in the same manner as any CVNC command.
- Check macros contain a set of procedures for a CVNC command, which are executed after the command is processed. They check to see that the executed command meets certain criteria. For example, a check macro could verify that the spindle retracts from the part before indexing a rotary table.
- Output macros create customized output for CVNC by passing machine control statements directly to the output file.
- Point macros are accessed at predefined points within command execution. They can be used, for example, with the AREAMILL (CVNC-M2), SURFCUT3 (CVNC-M3), SURFCUT4 (CVNC-M4), and SURFCUT5 (CVNC-M5) commands. Point macros issue pass-through statements directly to the output file, allowing you to add machine control functions within tool paths.

CVNC supplies several command macros that are installed and ready to use. These macros provide valuable part programming capabilities. For example, one macro automatically face mills an area defined by two digitizes.

Please note: To write your own CVMAC macros, you must purchase the CVMAC compiler and linker.

For more information, see *Customizing CVNC*.

Output

CVNC generates an output file in the format of your choice from a JCF. You can generate output in CLFile, APT, and COMPACT II.

In addition to choosing the output file format, you can also

- Selectively turn output on or off to generate output for certain JCF blocks.
- Invoke an auditing facility that inserts JCF commands into the output file as comment lines. This is useful when debugging a JCF part program.
- Process output in the background. This means that output processing can be done separately from a CVNC/CADDS session; it will not interfere with your CVNC or CADDS session.
- Generate output from a previously processed JCF without entering CADDS. This is done with a OS level command.
- Use CVNC pass-through statements that contain CLFile, APT, or COMPACT II statements. Pass-through statements are used like CVNC commands in your JCF, but they do not affect CVNC processing. For example, the following CLFile/APT language pass-through statement specifies a machine tool:

```
MACHIN 'BABG01'
```

- Write output macros that contain customized output for CVNC nonmotion commands.

The figure on the next page shows a JCF. The three remaining figures show JCF output in the three available formats.

For more information, see the *CVNC System User Guide and Menu Reference*.

Figure 2-12 Sample JCF

```
[JCF  CVNC MILL HEADER LINE           6-26-92 13:20
CADDSTATION ]
OUTPUT CLFILE TEXT 'CLFILEEXAMPLE'
OUTPUT APT 'APTEXAMPLE'
OUTPUT COMPACT 'COMPACTEXAMPLE'
SELECT LAY 10
DEFTOOL 'MILL2-1' MILL DIA 1
HOMEPT FROM X -3 Y 1 Z 5;
CLEAR HOME
CHGTOOL 3 'MILL2-1'
SPEED RPM 1800 CLW
FEED APPROACH IPM 80
FEED CLEAR RAPID
FEED CONNECT IPM 20
FEED CUT IPM 20
FEED PLUNGE IPM 12
FEED RETRACT IPM 40
COOLANT MIST
PLANE ZWORK 0 ZREF
AREAMILL LACE CUTDIR XAXIS BOUND LOC X -4 Y -4 X 4 Y
-4 X 4 Y 2 X -4 Y 2;
RETRACT
COOLANT OFF
CLEAR HOME
[DONE]
```

These are three possible OUTPUT commands. Only one of these command lines appears in the actual JCF. Which one depends on the output type specified.

CLfile Output

```
FROM -3.0000, 1.0000, 5.0000
RAPID
GOTO -3.0000, 1.0000, 5.0000
LOADTL 3 LENGTH 0.0000
SPINDL 1800.0000 RPM CLW
FEDRAT 20.0000 IPM
COOLNT MIST ON
RAPID
GOTO -3.0000, 1.0000, 15.0000
GOTO -3.5000, 1.5000, 15.0000
FEDRAT 80.0000 IPM
```



```
GOTO -3.5000, 1.5000, 10.0000
FEDRAT 12.0000 IPM
GOTO -3.5000, 1.5000, 0.0000
FEDRAT 20.0000 IPM
GOTO 3.5000, 1.5000, 0.0000
GOTO 3.5000, 0.7500, 0.0000
GOTO -3.5000, 0.7500, 0.0000
GOTO -3.5000, 0.0000, 0.0000
GOTO 3.5000, 0.0000, 0.0000
GOTO 3.5000, -0.7500, 0.0000
GOTO -3.5000, -0.7500, 0.0000
GOTO -3.5000, -1.5000, 0.0000
GOTO 3.5000, -1.5000, 0.0000
GOTO 3.5000, -2.2500, 0.0000
GOTO -3.5000, -2.2500, 0.0000
GOTO -3.5000, -3.0000, 0.0000
GOTO 3.5000, -3.0000, 0.0000
GOTO 3.5000, -3.5000, 0.0000
GOTO -3.5000, -3.5000, 0.0000
GOTO 3.5000, -3.0000, 0.0000
FEDRAT 40.0000 IPM
GOTO 3.5000, -3.0000, 10.0000
COOLNT OFF
RAPID
GOTO -3.0000, 1.0000, 5.0000
```

APT Output

```
FROM/ -3.0000, 1.0000, 5.0000
RAPID
GOTO/ -3.0000, 1.0000, 5.0000
LOADTL/ 3, LENGTH, 0.0000
SPINDL/ RPM, 1800.0000, CLW
FEDRAT/ 20.0000, IPM
COOLNT/ MIST
RAPID
GOTO/ -3.0000, 1.0000, 15.0000
GOTO/ -3.5000, 1.5000, 15.0000
FEDRAT/ 80.0000, IPM
GOTO/ -3.5000, 1.5000, 10.0000
FEDRAT/ 12.0000, IPM
GOTO/ -3.5000, 1.5000, 0.0000
FEDRAT/ 20.0000, IPM
GOTO/ 3.5000, 1.5000, 0.0000
GOTO/ 3.5000, 0.7500, 0.0000
GOTO/ -3.5000, 0.7500, 0.0000
GOTO/ -3.5000, 0.0000, 0.0000
GOTO/ 3.5000, 0.0000, 0.0000
GOTO/ 3.5000, -0.7500, 0.0000
GOTO/ -3.5000, -0.7500, 0.0000
GOTO/ -3.5000, -1.5000, 0.0000
GOTO/ 3.5000, -1.5000, 0.0000
GOTO/ 3.5000, -2.2500, 0.0000
```

```
GOTO/   -3.5000,   -2.2500,   0.0000
GOTO/   -3.5000,   -3.0000,   0.0000
GOTO/    3.5000,   -3.0000,   0.0000
GOTO/    3.5000,   -3.5000,   0.0000
GOTO/   -3.5000,   -3.5000,   0.0000
GOTO/    3.5000,   -3.0000,   0.0000
FEDRAT/  40.0000,IPM
GOTO/    3.5000,   -3.0000,   10.0000
COOLNT/  OFF
RAPID
GOTO/   -3.0000,    1.0000,    5.0000
```

COMPACT II Output

```
BASE, XA0.0000, YA0.0000, ZA0.0000
MOVE, -3.0000XB,  1.0000YB,  5.0000ZB
MOVE, -3.0000XB,  1.0000YB,  5.0000ZB
ATCHG, TOOL 3, GL 0.0000, TD 1.0000
MOVE, -3.0000XB,  1.0000YB,  15.0000ZB, RPM 1800.0000, CON2
MOVE, -3.5000XB,  1.5000YB,  15.0000ZB
CUT,  -3.5000XB,  1.5000YB,  10.0000ZB, IPM 80.0000
CUT,  -3.5000XB,  1.5000YB,  0.0000ZB, IPM 12.0000
CUT,   3.5000XB,  1.5000YB,  0.0000ZB, IPM 20.0000
CUT,   3.5000XB,  0.7500YB,  0.0000ZB
CUT,  -3.5000XB,  0.7500YB,  0.0000ZB
CUT,  -3.5000XB,  0.0000YB,  0.0000ZB
CUT,   3.5000XB,  0.0000YB,  0.0000ZB
CUT,   3.5000XB, -0.7500YB,  0.0000ZB
CUT,  -3.5000XB, -0.7500YB,  0.0000ZB
CUT,  -3.5000XB, -1.5000YB,  0.0000ZB
CUT,   3.5000XB, -1.5000YB,  0.0000ZB
CUT,   3.5000XB, -2.2500YB,  0.0000ZB
CUT,  -3.5000XB, -2.2500YB,  0.0000ZB
CUT,  -3.5000XB, -3.0000YB,  0.0000ZB
CUT,   3.5000XB, -3.0000YB,  0.0000ZB
CUT,   3.5000XB, -3.5000YB,  0.0000ZB
CUT,  -3.5000XB, -3.5000YB,  0.0000ZB
CUT,   3.5000XB, -3.0000YB,  0.0000ZB
CUT,   3.5000XB, -3.0000YB,  10.0000ZB, IPM 40.0000
MOVE, -3.0000XB,  1.0000YB,  5.0000ZB, COF
```

CVNC Application Summaries

In addition to the features already described, some features are specific to the six CVNC applications. They provide part programming capabilities and solutions for different manufacturing applications.

CVNC-M2

CVNC-M2 provides

- 2- and 2 1/2-axis tool motion relative to CADDs construction planes
- 4- and 5-axis positioning
- Rotary tables/heads
- Hole processing

Tool motion can be performed relative to any combination of lines, arcs, conics, B-splines, Nsplines, Cpoles, and strings.

CVNC-M2 provides 2- and 2 1/2-axis support for mills/drills, 3-axis machining centers, 4-axis machining centers, 4-axis boring mills, and 5-axis machining centers.

CVNC-M2 also provides basic support for 2-axis wire electric discharge machines (EDM), and 2-axis flame and laser cutters.

CVNC-M3

CVNC-M3 provides 3-axis surface machining of any combination of CADDs model geometry, including

- Conventional surfaces and surfaces of revolution (RSURFS, BSURFS, SREVs, etc.)
- Advanced Surface Design (ASD) surfaces
- Nonuniform Rational B-spline Surfaces (NSURFS/NURBS)
- Solids and surfaces on solids, including trimmed boundaries (TSURFs)
- Three-dimensional contouring of any combination of lines, arcs, conics, B-splines, Nsplines, Cpoles, and strings.

Three-axis surface machining can be performed with fixed tool orientation normal to any CADDs construction plane using 7-parameter tools, including ball end mills, end mills with corner radii, and flat-bottom end mills.

CADDs wireframe geometry can be used for 2- and 2 1/2-axis CVNC-M2 functions while in CVNC-M3.

CVNC-M5

CVNC-M5 provides 4- and 5-axis surface machining of any combination of CADDs model geometry, including

- Conventional surfaces (RSURFS, BSURFS, etc.)
- Advanced Surface Design (ASD) surfaces
- Nonuniform Rational B-spline Surfaces (NSURFS/NURBS)
- Surfaces of revolution (SREVs)
- Solids and surfaces on solids, including trimmed boundaries (TSURFs), with some limitations

Tool motion can be performed relative to any combination of lines, arcs, conics, B-splines, Nsplines, Cpoles, and strings.

CVNC-M5 supports all classes of NC mills and machining centers. It provides sophisticated tool axis controls for surface machining, surface intersection machining, and swarfcutting operations. Profiling with full 5-axis control is also included.

CVNC-T2

CVNC-T2 provides

- 2-axis continuous path profiling
- Rough turning/boring/facing at any angle
- 2-axis positioning
- Twin turret support
- Centerline hole processing support

Tool motion can be performed relative to any combination of lines, arcs, conics, B-splines, Nsplines, Cpoles, and strings. CVNC-T2 supports front or rear single or

twin turret 2-axis turning equipment that has horizontal or vertical spindle orientation.

CVNC-T2 supports twin turrets with independent slides; however, only one turret is displayed and programmed at a time.

CVNC-P2

CVNC-P2 provides 2-axis punching and nibbling. CVNC-P2 supports turret presses with indexable turret stations.

CVNC-P2 supports eight standard punch shapes. Attributes of these shapes are used to calculate nibbling, recognize boundaries, perform tonnage calculations, and calculate area clearance.

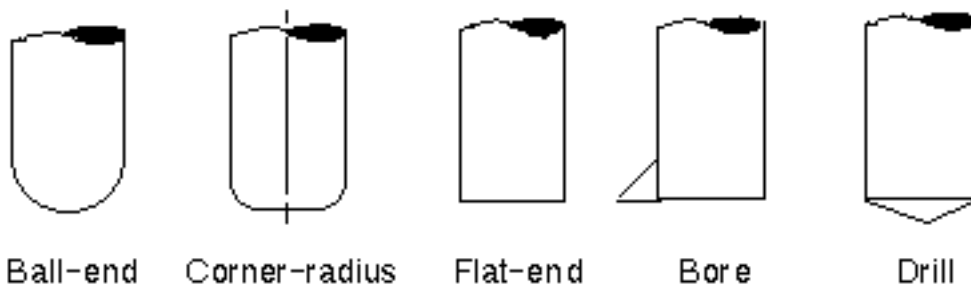
CVNC-M2

CVNC-M2 features provide a part programming system for 2 1/2-axis tool motion, hole processing, and 4- and 5-axis positioning. In addition to the base system features, the main features of CVNC-M2 are

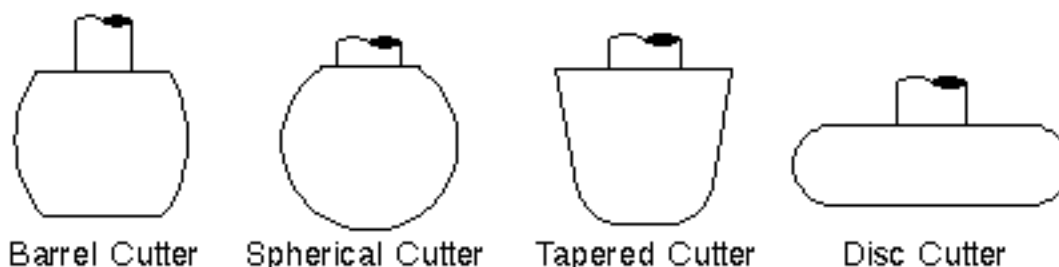
- Tool shape definition
- Configuration
- Discrete motion
- Depth control
- Stock
- Boundary profiling
- Pocketing with islands
- Lace pocketing with islands
- Hole processing

Tool Shape Definition

CVNC-M2 supports the following tools: end mills with or without corner radii, drills, taps, boring tools, and center drills. Tools are defined with a tool type, dimensions, and a 22-character description field.



In addition, surface machining commands within CVNC-M3 support 7-parameter mill tool definitions. Examples are shown below.



Configuration

You can define the machine tool coordinate system and its orientation with respect to the CADDs model coordinate system.

You can also define the machine's rotary devices. You can configure machines with a combination of rotating tables and/or heads. It is also possible to specify the range of motion of each axis of the target machine tool.

Discrete Motion

Commands are provided for absolute and incremental motion. Discrete motion is relative to wireframe entities with options allowing individual axes suppression, feed rate, and depth control.

Depth Control

You can define retract, clear, approach, and work planes relative to model entities or absolute and incremental coordinates. Each of these planes has its own positioning commands with associated feed rate control that allow you to easily control the depth of the tool.

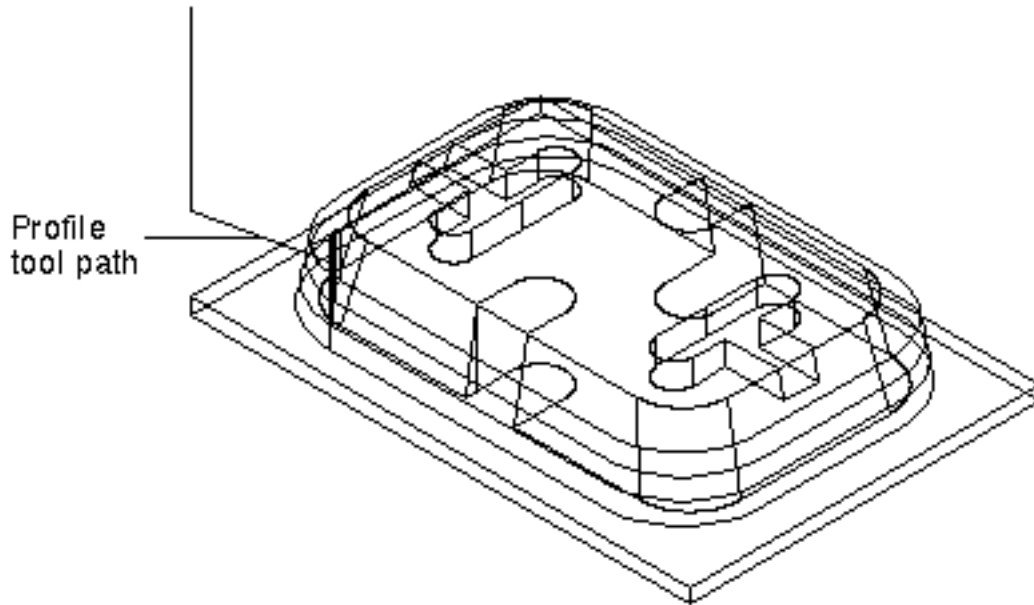
Stock

You can specify stock offsets (thicknesses) for boundaries to be pocketed, lace pocketed, profiled, or cut. Offsets can be associated with individual entities.

Boundary Profiling

CVNC-M2 provides continuous profiling of up to 512 CADDs wireframe entities, with interference analysis and circular arc approximation. You can select the model entities via getdata or with one or more named entity groups.

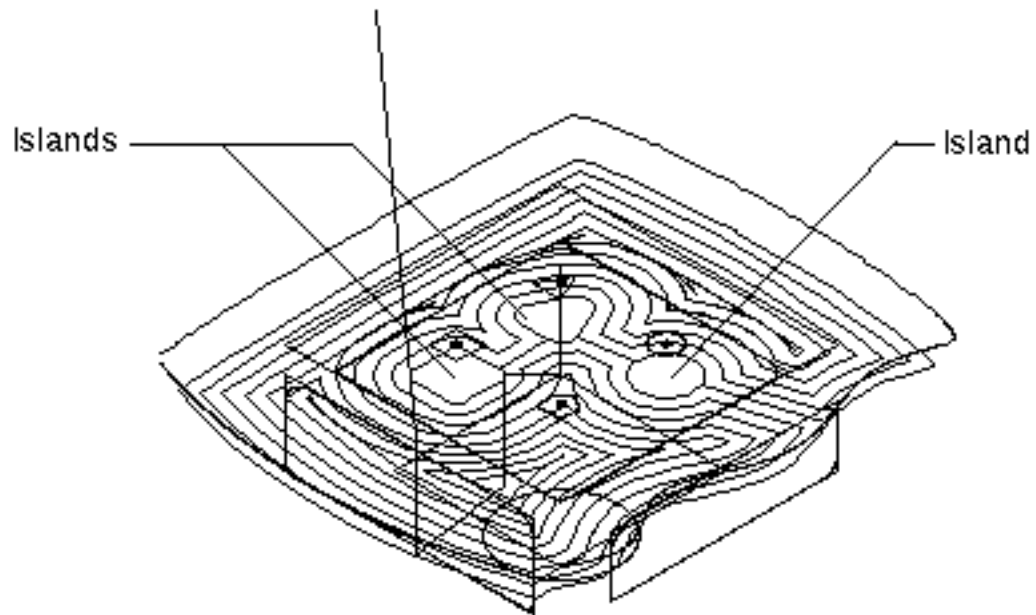
CVNC-M2 includes a macro for multiple depth, side, or draft angle profiling cuts with an unlimited number of passes.



Pocketing with Islands

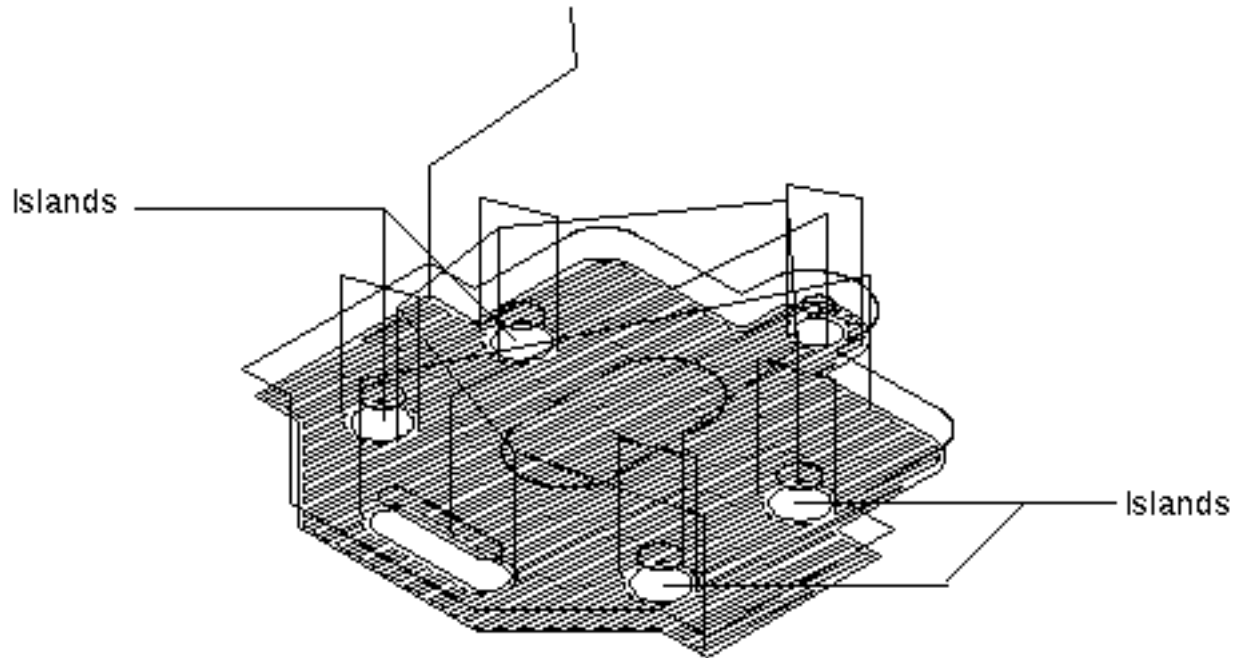
The POCKET command performs area removal within a closed boundary of up to 512 CADDs wireframe entities. Pocketing can be done with either ramp or plunge entry and full feed rate and depth control. You can select the model entities via getdata or with one or more named entity groups.

You can define up to 20 interior islands (using entity groups or digitizes) with no entity restrictions. Pocketing includes CVMAC macro-controlled multiple depth cuts with optional draft angle and an unlimited number of passes.



Lace Pocketing with Islands

A lace pocketing command (AREAMILL) performs straight line bidirectional or unidirectional tool motion for removing material within a wireframe boundary. You can define up to 50 islands within this boundary.



Hole Processing

CVNC-M2 provides drill, bore, tap, and countersink commands with optional full cycle emulation for the first location or all locations where holes are machined. You can display default tools or user-defined subfigures.

You can generate avoidance moves for positioning over obstacles, such as clamps or fixtures. The avoidance area can be defined incrementally or in reference to CVNC depth control planes.

Forward and reverse options provide a convenient method for generating efficient hole processing motion, which can also be optimized to generate the shortest path between locations.

For more information, see the *CVNC-M2 User Guide*.

CVNC-M3

CVNC-M3 features provide a part programming system for 3-axis surface machining. In addition to base system and CVNC-M2 features, CVNC-M3 includes

- Integration with CVNC-M2
- Straight line, parametric, and control curve machining over multiple surfaces
- Constant z-height profile finishing over multiple surfaces
- Containment of straight line, parametric, and control curve tool paths
- Surface intersection machining (including part and drive surface ordering)
- 3-D projection of 2-D tool paths
- Collection of surfaces and containment boundaries into surface lists
- Collision checking
- Approach and retract control
- Scallop height and stepover control
- Surface extensions at the start and end of each pass, and extra passes before the first pass and after the last
- 3-D contour machining
- 7-parameter tool support
- Material modeling for rough machining

Integration with CVNC-M2

All CVNC-M2 mill/drill functions, 2-D discrete motion, 2-D profiling, lace cutting, and pocketing motion are fully integrated and available with CVNC-M3.

JCFs written in CVNC-M2 can be accessed and executed within CVNC-M3.

Straight Line Machining

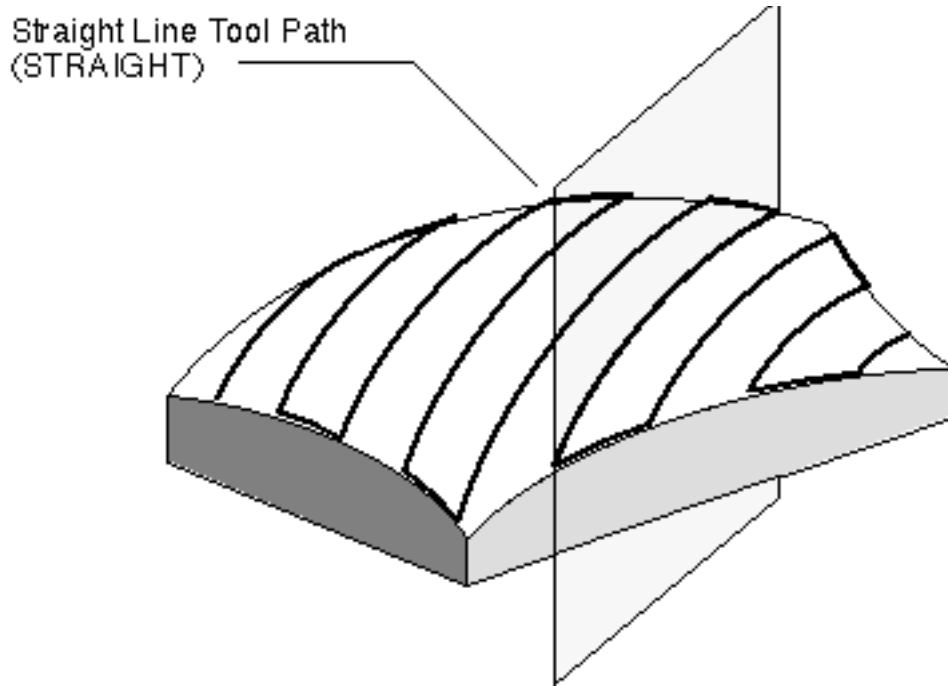
CVNC-M3 provides degouged straight line cutting capabilities on

- Surfaces (including surfaces on solids and trimmed surfaces)
- Surface lists

Surfaces can overlap (the uppermost surface is machined). Neighboring surfaces can have gaps between them.

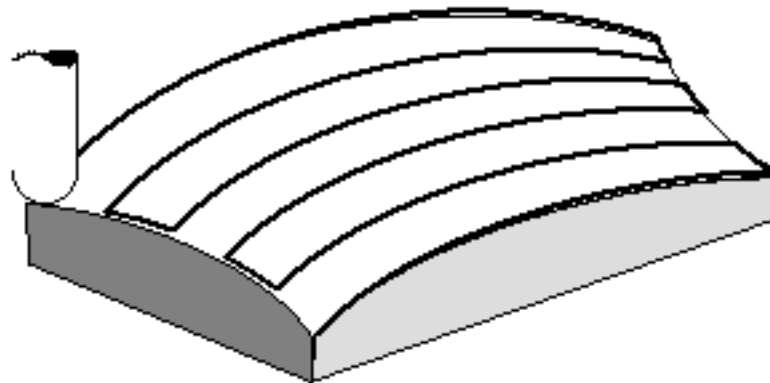
During straight line machining, the tool tip follows a straight line across the surface regardless of the surface topology; a series of parallel vertical planes are projected down to the surface (in the active construction plane).

The point of intersection between the plane and the surface forms the tool path.



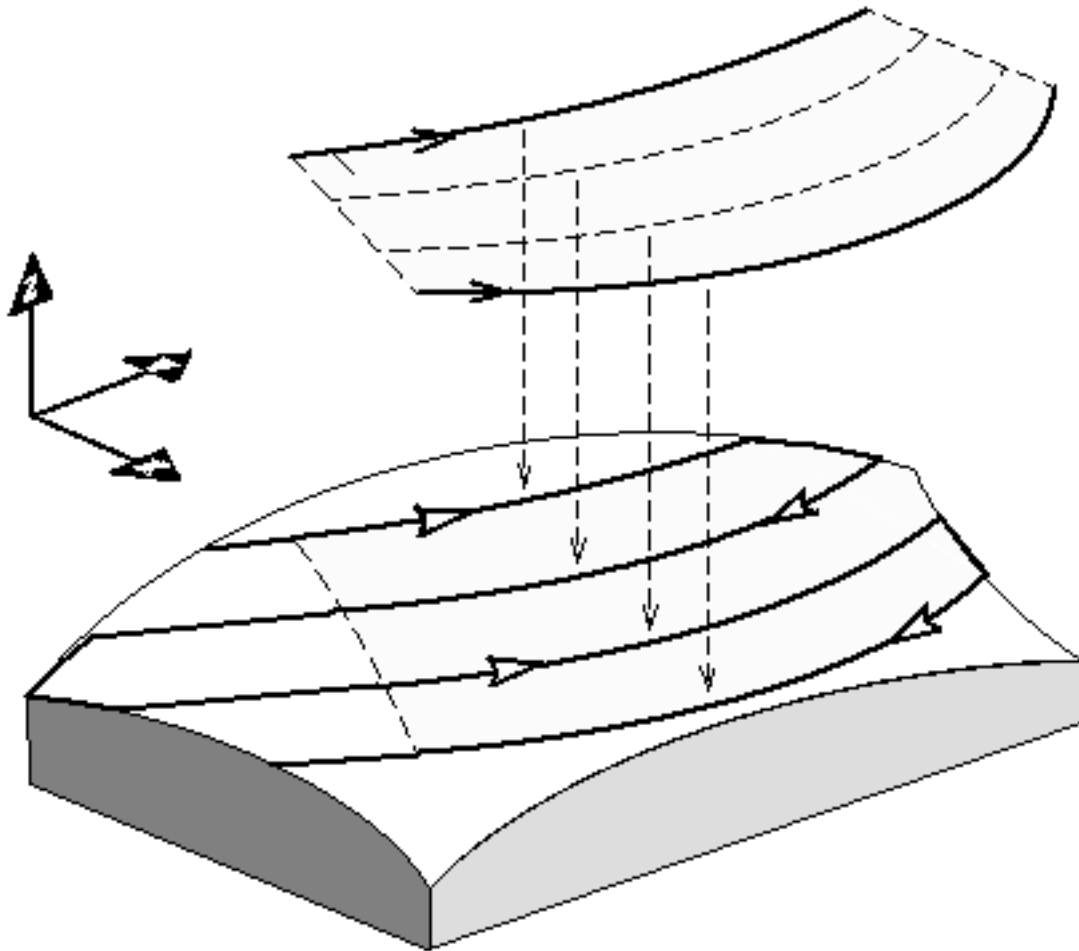
Parametric Machining

CVNC-M3 provides parametric machining, driving the contact point of the tool along the isoparametric lines of a surface or surface list. Parametric machining may be in either bidirectional or unidirectional mode.



Control Curve Machining

CVNC-M3 provides control curve machining, whereby the tool path is formed by projecting a series of control curves onto the designated part surfaces.



Constant Z-Height Profile Finishing

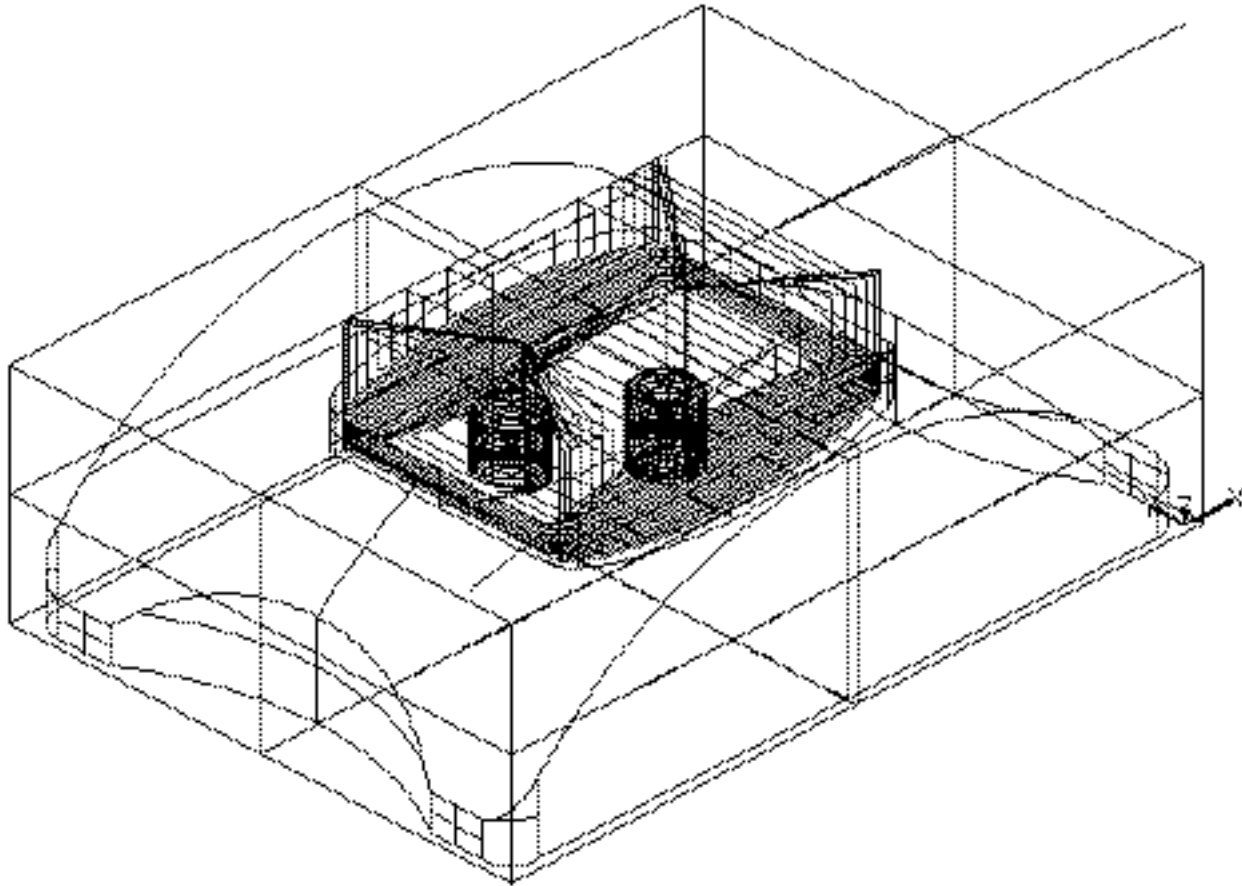
Companion to SURFCUT3, ZPROF3 provides profile finishing capability, particularly for vertical surface geometry.

Profile finishing (also referred to as waterline profiling) produces a series of profile cuts around part surfaces.

Tool path output consists of a series of horizontal profiles (constant z) connected by various methods.

You can control the minimum and maximum z values of cuts to be output. In addition, you can specify the stepover between cuts and the starting location for each profile.

ZPROF3 machines the inside of this part cavity:



With ZPROF3,

- The tool tip remains at a constant z height, resulting in the surface contact point varying in z according to the surfaces being machined.
- The stepover between cuts remains constant.
- You can specify any surface type and tool shape.
- You can select among several leadin/leadout motion and tool path connection options.

On near-vertical surfaces, ZPROF3 provides a good surface finish with a controlled number of cuts.

Uncut Area Marking

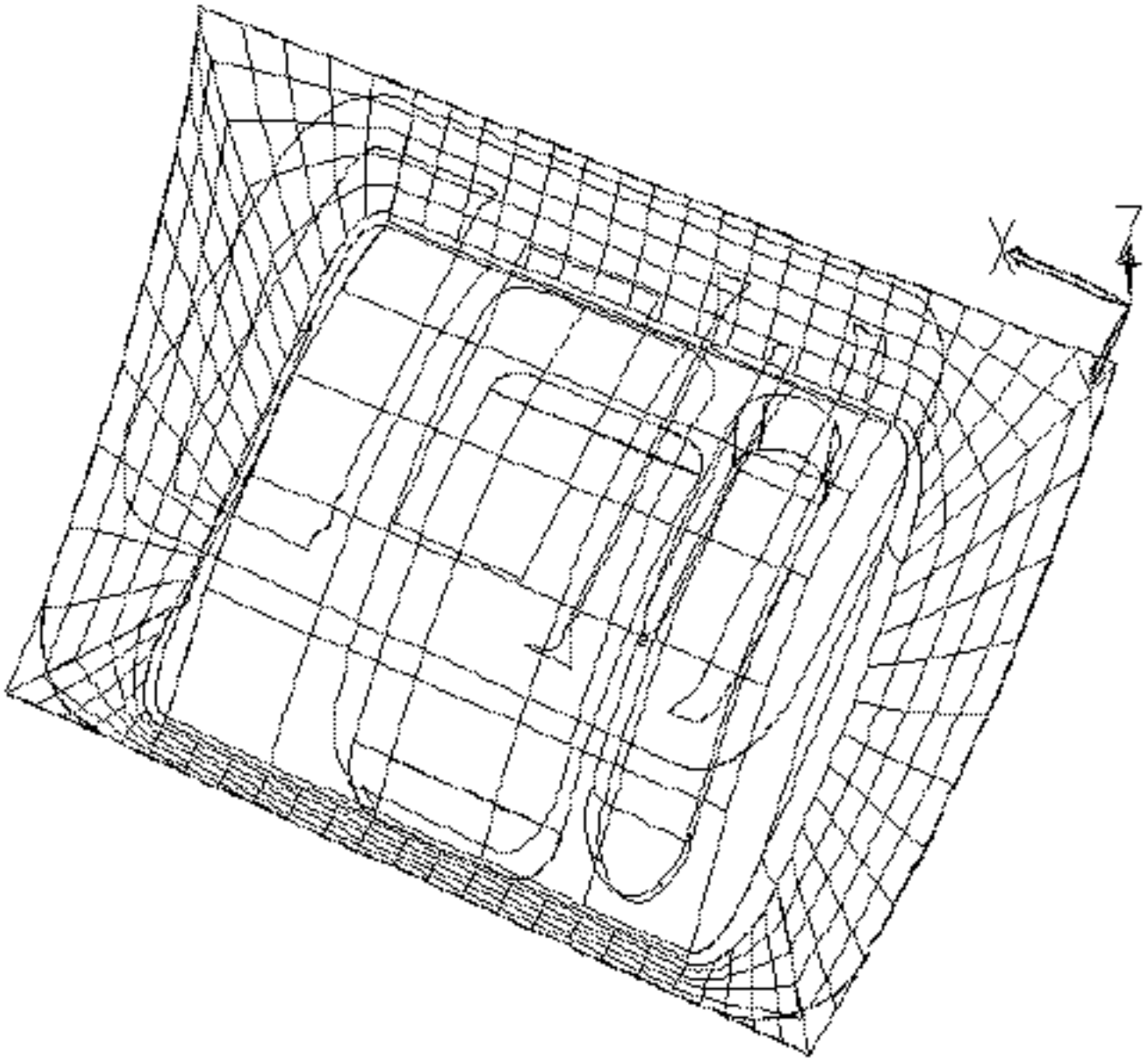
As SURFCUT3 creates the tool path, you can simultaneously create closed boundaries around any uncut area and display them as Nsplines.

You can subsequently machine these Nsplines

- Using SURFCUT3 with a different tool
- With other CVNC machining commands

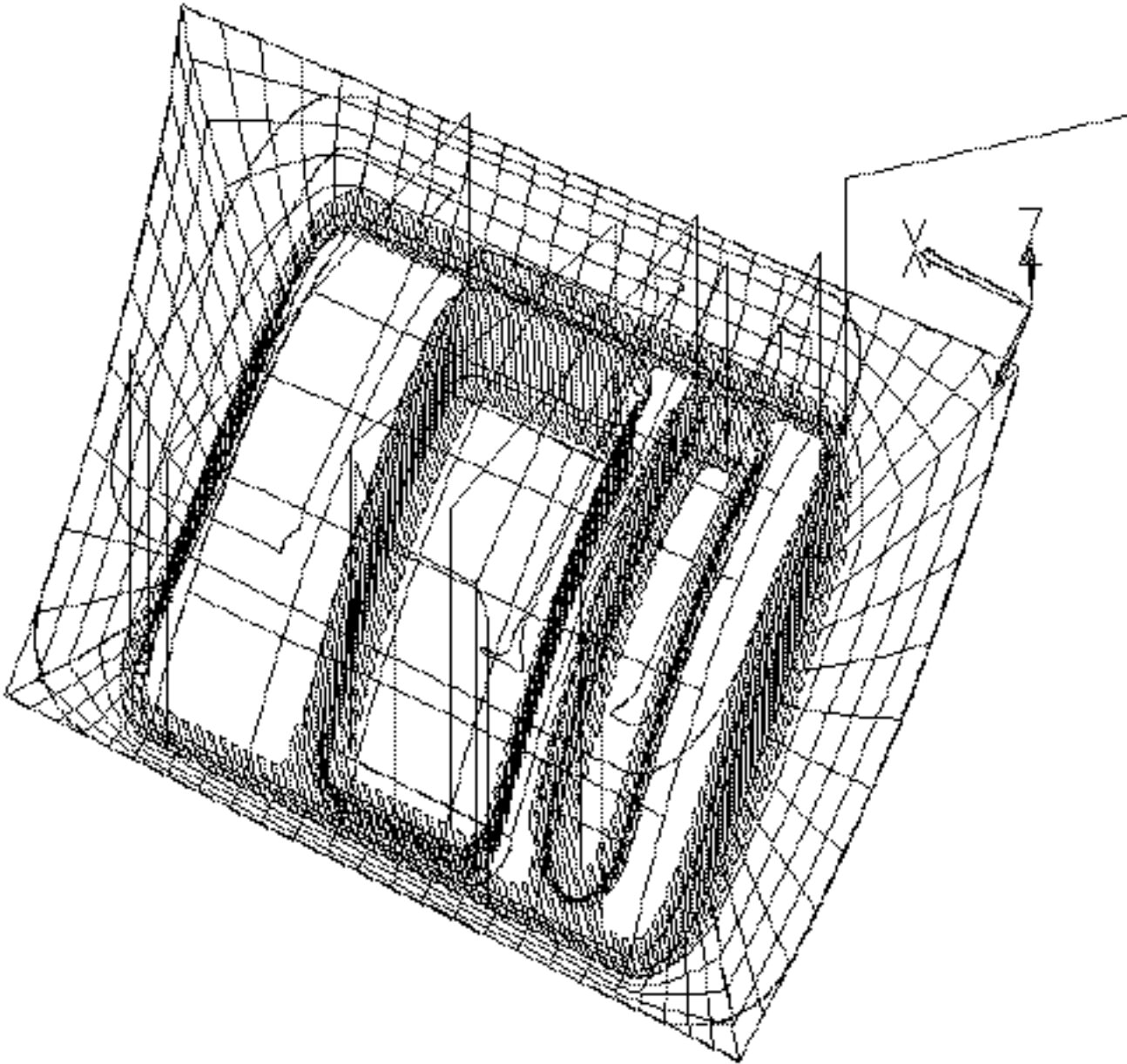
For example, SURFCUT3 machines the part shown in Figure 2-13, “SURFCUT3 with UNCUT Turned ON,” on page 2-44. By setting UNCUT on, the uncut area Nsplines are also created at the same time the tool path is being generated.

Figure 2-13 SURFCUT3 with UNCUT Turned ON



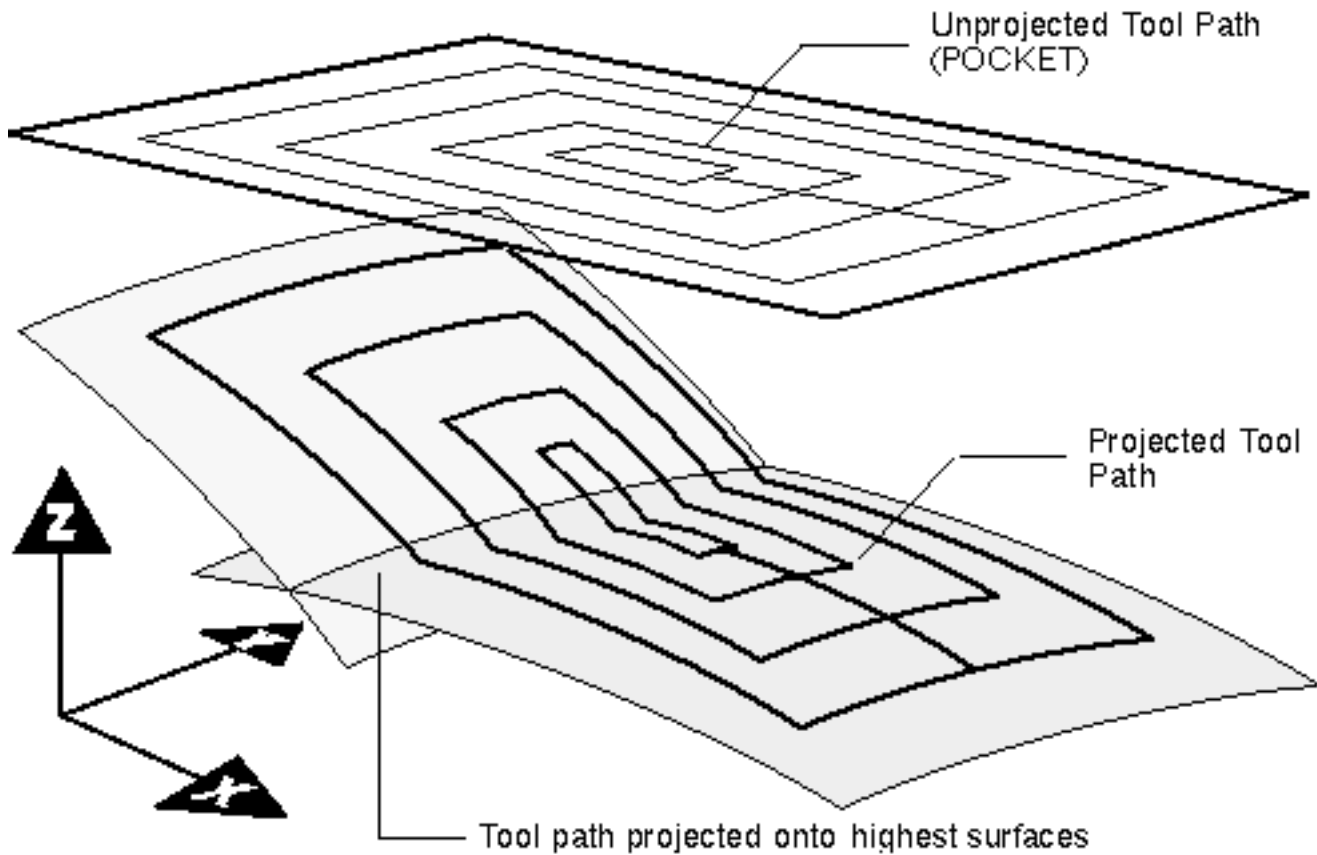
Next, AREAMILL machines the uncut regions, using the outer Nspline for the boundary and the 5 inner Nsplines as islands. Because projection mode is on, the AREAMILL is projected onto the part surfaces.

Figure 2-14 Projected AREAMILL of the UNCUT Areas



3-D Projection of 2-D Tool Paths

2 1/2-axis tool motion can be projected onto surfaces or surface lists. The equivalent 3-D motion is generated for the 2-D motion commands.



Surface Lists

You can group and name a number of surfaces or wireframe geometry into a surface list. The named group can be referenced in subsequent commands, such as

- Surface intersection machining
- Collision detection
- Containment generation

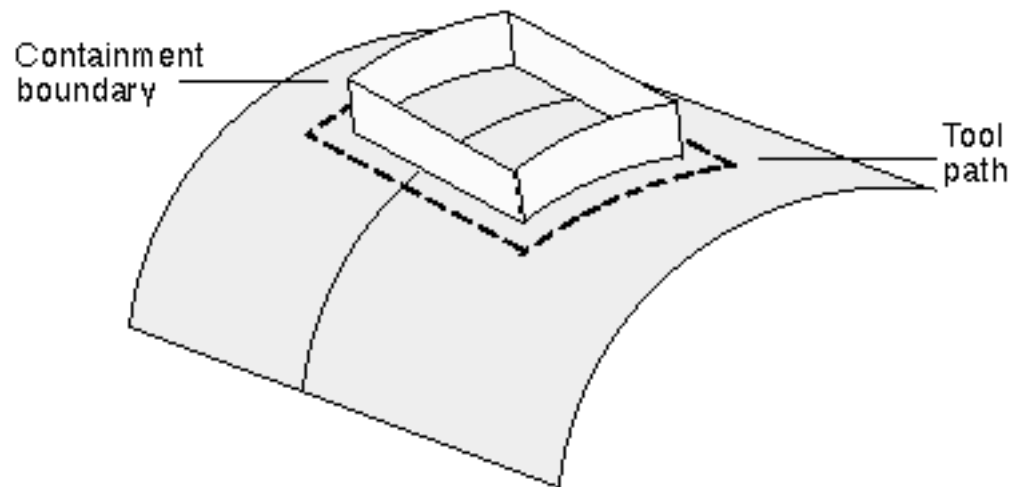
Surface lists can be

- Interactively edited
- Displayed with highlighting features
- Appended with previously defined lists

Containment

With containment facilities, you can specify the desired motion when the tool comes into contact with a boundary during straight line, parametric, or control curve machining.

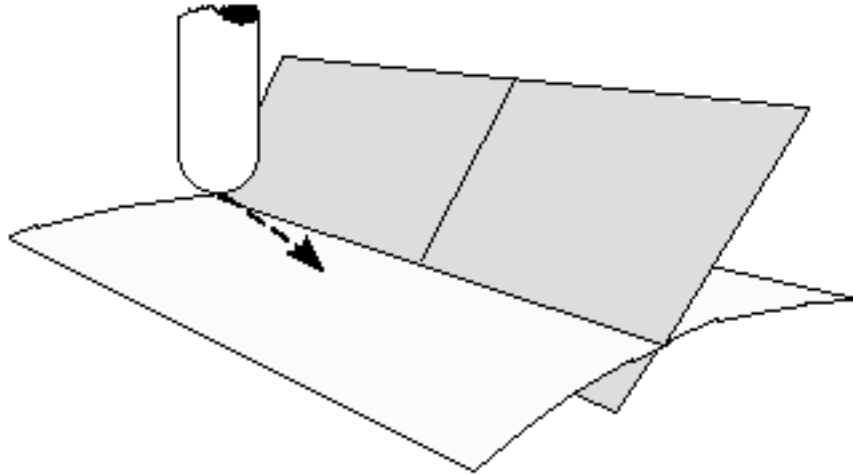
Four containment generation types are available, including wireframe. Parametric machining offers a fifth containment type that restricts machining to within specified rectangular parametric boundaries.



Surface Intersection Machining

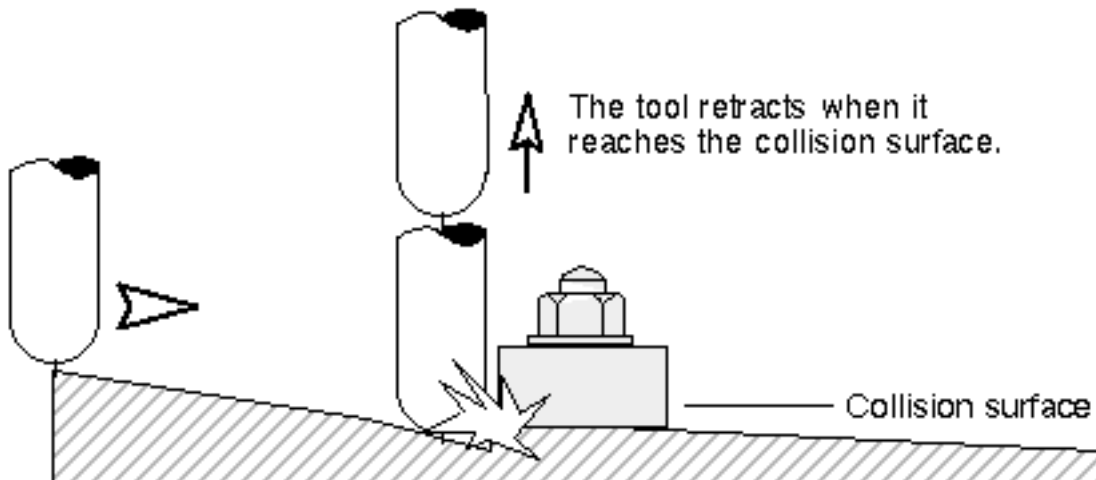
The tool can be driven along the intersection between a drive surface list and a part surface list.

You can start or stop the tool against a third surface, a drive or part surface edge, or at a defined proximity point. Drive surfaces can be supplied as projected wireframe geometry.



Collision Checking

CVNC-M3 uses tool and surface geometry to provide automatic collision checking that can be turned on or off. Collision checking detects and removes tool movement that would cause the tool to collide with specified collision surfaces.



Approach and Retract Control

You can define the direction and amount of approach and retract motion. This provides incremental moves to and from the surface extension points under associated feed rate control.

Scallop Height and Stepover Control

During parametric, straight line and control curve machining, you can specify the distance between cutting paths. This is called the stepover distance.

You can also specify the maximum height allowable for the peak of the surface area not cut between tool paths. This is called the scallop height.

Surface Extensions

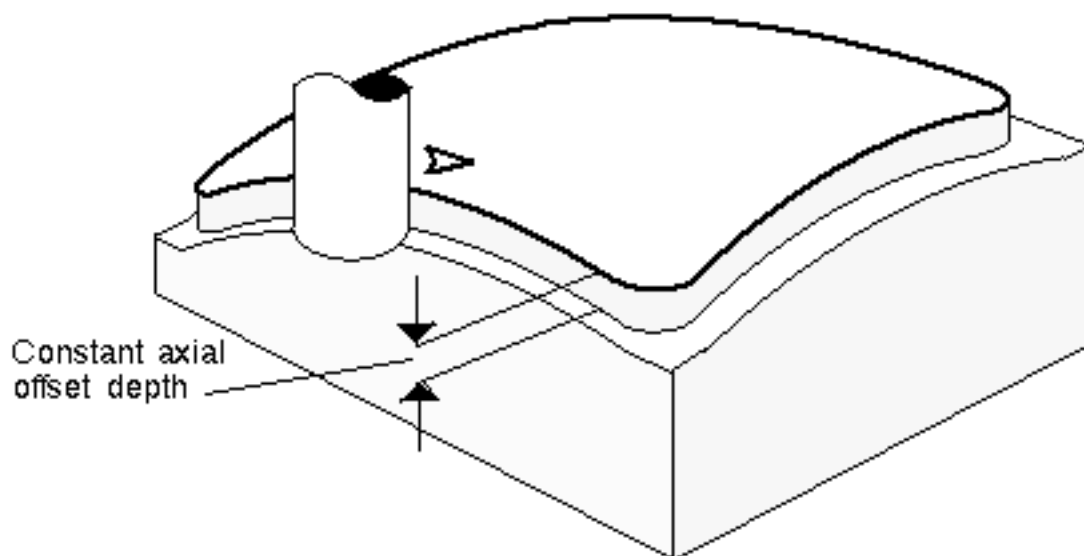
Bidirectional (LACE) and unidirectional (NOLACE) tool paths can be extended in the cutting direction at the first and/or last surface(s) to provide machining of rough material surrounding the model geometry. Extra passes can be added before the first surface cutting pass and after the last. This minimizes the need for construction geometry. Surface intersection tool paths, projected tool paths, and 3-D profiling tool paths can also be extended.

7-Parameter Tools

Within CVNC-M3 milling operations, you can define a 7-parameter tool using the MILL7 modifier of the DEFTOOL (Define tool) command.

Three-dimensional Profiling

CVNC-M3 offers 3-D profiling, whereby the tool remains at a constant depth relative to the profile curve geometry.



Material Modeling for Roughing

CVNC-M3 provides a rough machining capability that allows you to specify the workpiece and finished part geometry, and carry out waterline machining and access hole drilling operations.

For more information, see the *CVNC-M3 User Guide*.

CVNC-M5

CVNC-M5 provides a part programming system for 4- and 5-axis surface machining. In addition to the base system, CVNC-M2, and CVNC-M3 features, the main features of CVNC-M5 are

- Integration with CVNC-M2 and CVNC-M3
- Gouge avoidance
- Scallop height and stepover control
- Swarfcutting
- Profiling
- Fixed or variable lead and slew control
- Four-axis parametric machining

Integration with CVNC-M2 and M3

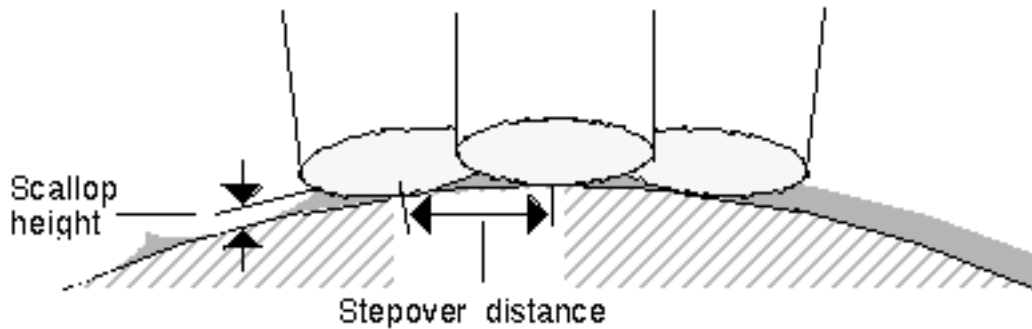
All 2- and 2 1/2-axis mill/drill functions, 2-D discrete motion, and 2-D profiling and pocketing motion are available with CVNC-M5. CVNC-M3 functions are also available, providing 3-axis surface machining on any CADDs model geometry, including NURBS and solid model surfaces.

Gouge Avoidance

Gouge avoidance is achieved automatically during surface machining for surfaces that are tangent or nontangent in the direction of the cut. This is achieved by varying the lead angle to avoid obstacles either in front of or behind the tool.

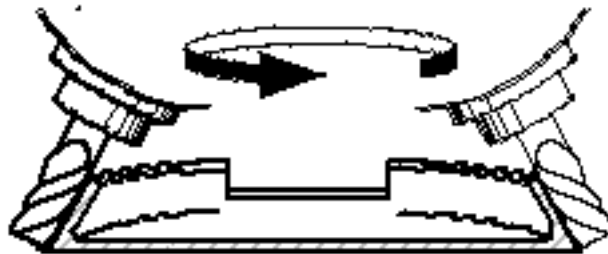
Scallop Height and Stepover Control

During parametric, straight line and control curve machining, you can specify a chordal scallop height value. This value is calculated by adjusting the tool orientation or the stepover between passes or both to achieve a true scallop height.



Swarfcutting

CVNC-M5 supports swarfcutting (also known as peripheral or slab milling), which machines with the tool side. Swarfcutting reduces the number of setups needed to machine a component and removes more material on a single pass than with conventional 5-axis bottom cutting.



Profiling

Using the CVNC-M5 profiling command - PROFILE5, you can perform boundary machining using the cutter side with full 5-axis control. PROFILE5 is especially useful for slab milling models designed with solids and trimmed surfaces. You can define the machining boundary by either

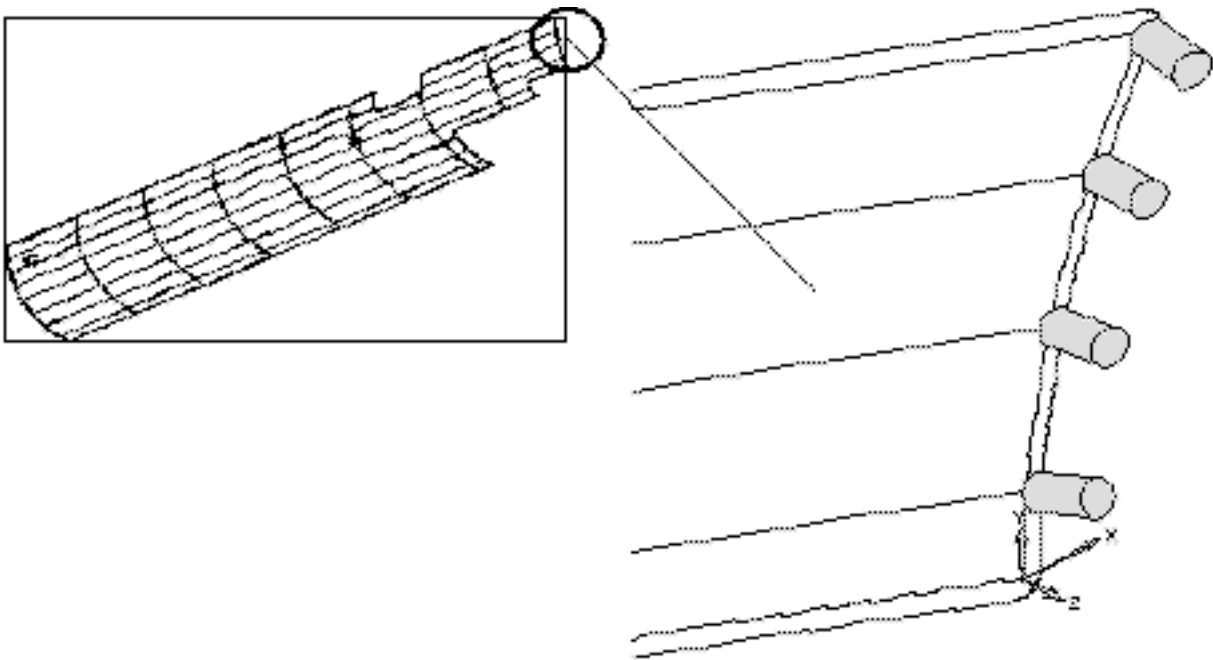
- Wireframe geometry
- The perimeter of a surface group

During machining, the tool can remain on or tangent to the boundary. You can specify the cut depths, with respect to the boundary. You can also specify tool axis fanning.

You can control the tool axis

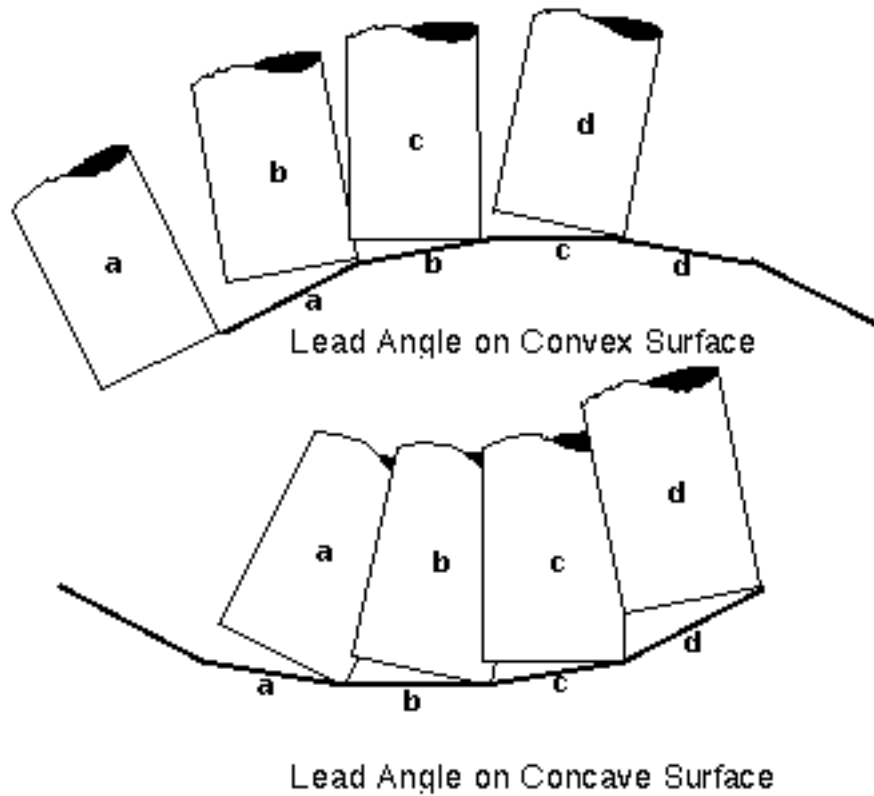
- By specifying two or more orientations, one at the boundary start and another at the end (with others optionally in between). CVNC interpolates the tool axes from this information.
- Where the boundary is formed by the perimeter of a surface group, via the surfaces normals along the boundary.

Here an an illustration showing an outer edge profile and tool axis control normal to the part surface.



Fixed or Variable Lead and Slew Control

During parametric, straight line and control curve machining, you can specify positioning of the tool tip at centerline or the face of the tool in the direction of the cut.



4-Axis Parametric Machining

CVNC-M5 provides 4-axis parametric machining (continuous machining in up to three linear and one rotational axis) and 4-axis surface intersection machining. Tool shapes can be the same as within 5-axis operations, including tapered tool support.

For more information, see the *CVNC-M5 User Guide*.

CVNC-T2

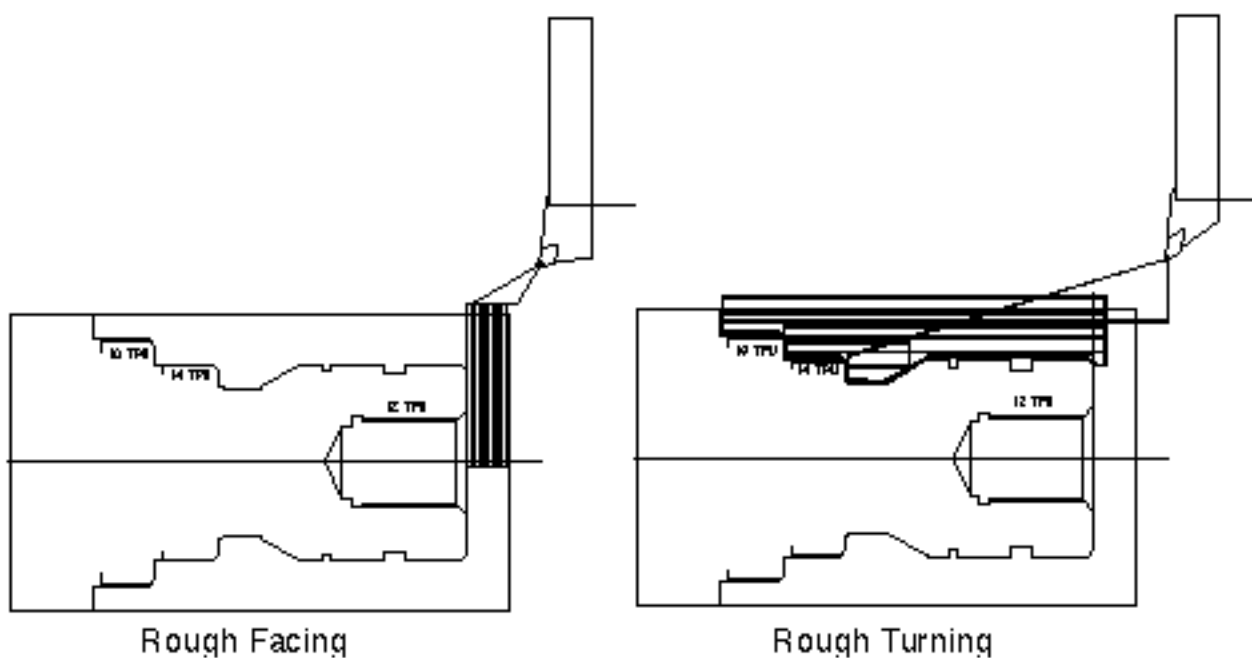
In addition to the base system features, the main features of CVNC-T2 are

- Rough turning, boring, and facing
- Boundary profiling
- Discrete motion
- Threading
- Grooving
- Configuration
- Tool shape definition
- Axis mirroring and transformation

Rough Turning, Boring, and Facing

A single command performs area removal for external, internal, or facing cuts at any slope between rough and finish boundaries. You can define a simple boundary or complex forged or cast material shapes.

Within an operation, you can control entry, cutting, retraction, and repositioning moves. You can define a minimum and maximum cutting depth for each pass, and then optimize the number of passes required within the defined cutting depths.



Boundary Profiling

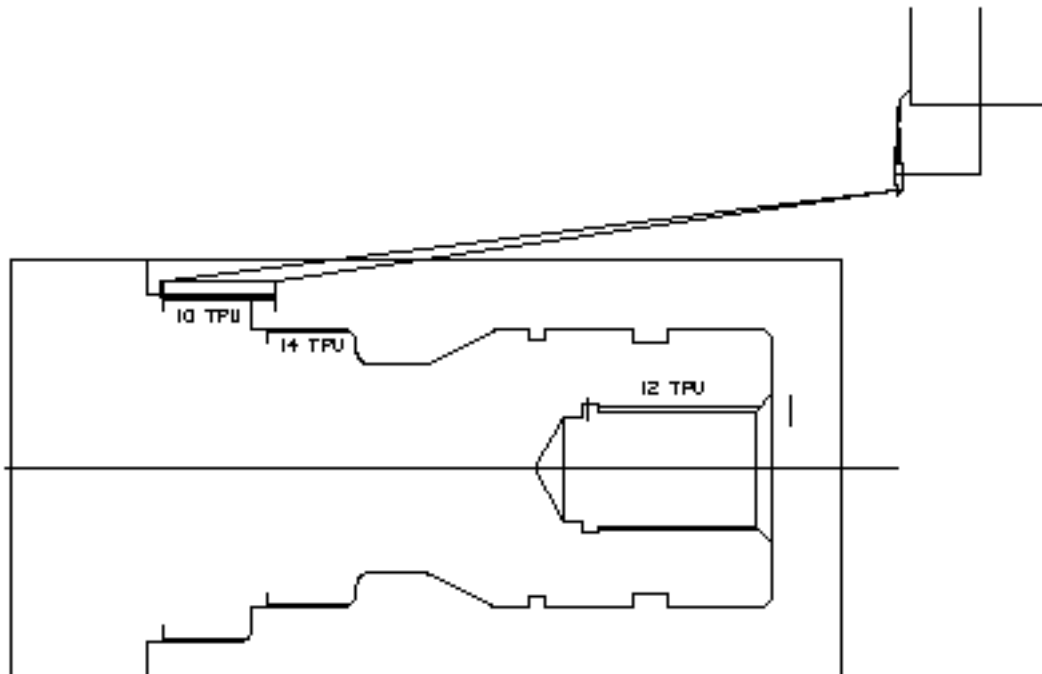
CVNC-T2 provides continuous profiling of up to 512 CADDSS wireframe entities, with interference analysis and circular arc approximation. You can select the model entities via `getdata` or with one or more named entity groups.

Discrete Motion

Commands are provided for convenient motion along any axes, absolute and incremental coordinate moves, or moves calculated relative to a mixture of wireframe entities with individual axes suppression, feed rate, and depth control. These commands allow for precise control of the tool between higher level commands, such as rough turning.

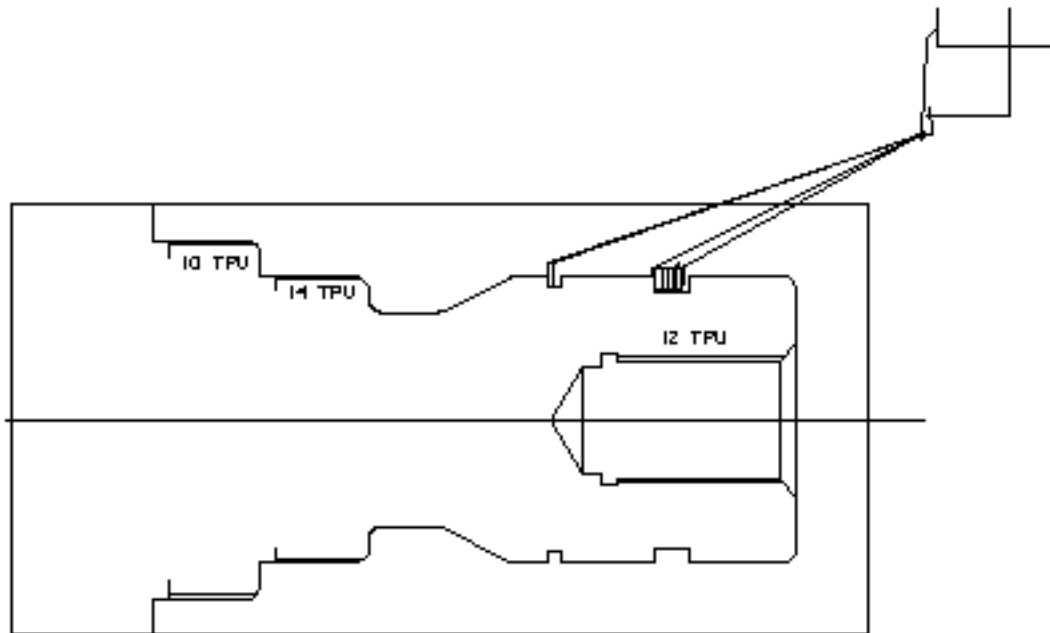
Threading

A single command generates all passes necessary to machine an external or internal, straight or tapered thread form. You can specify single or multiple start threads, the number of passes, the amount of depth per pass, lead angles, lead amount, and feed mode.



Grooving

A CVMAC macro is available to generate all passes necessary to rough (and optionally finish) a simple groove shape. You can specify that sharp corners be broken, and you can generate a single wipe pass to clean up the bottom of the groove.



Configuration

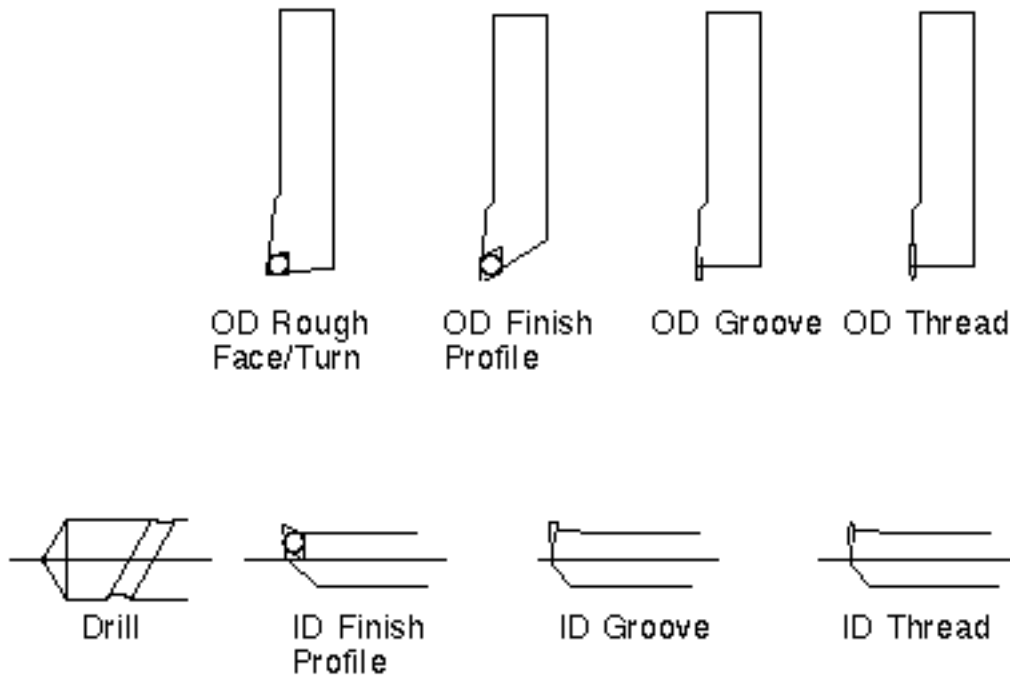
You can define the machine tool coordinate system and its orientation with respect to the CADDs model coordinate system.

You can also define the primary cross-slide quadrant on the machine tool and nominate the primary turret. Subsequent tool changes automatically generate appropriate postprocessor instructions and tool path display.

Tool Shape Definition

CVNC-T2 supports the following tools: external right or left hand turning, threading or grooving tools, internal boring threading or grooving tools, and

centerline drilling and tapping tools. Tools are defined with a tool type, dimensions, and a 22-character description field.



Axis Mirroring and Transformation

The SETAX (Set axis) command allows you to conveniently display motion for individual turrets in their respective machine coordinate systems, yet generate output in any coordinate system required for postprocessing.

For more information, see the *CVNC-T2 User Guide* and *CVNC-T2 Command Reference*.

CVNC-P2

The features of CVNC-P2 provide a part programming system for 2-axis punching and nibbling. In addition to the base system features, the main features of CVNC-P2 are

- Machine and turret definition
- Punch definition
- Variable angle nibbling and punching
- Sorting and optimizing punch hits
- Area clearance
- Configurable output
- Automatic search for best punch
- Repeating a block of punching operations

Machine and Turret Definition

You can define and modify machine and turret configurations. The coordinate system command (DATUM) identifies a construction plane used to establish machine zero.

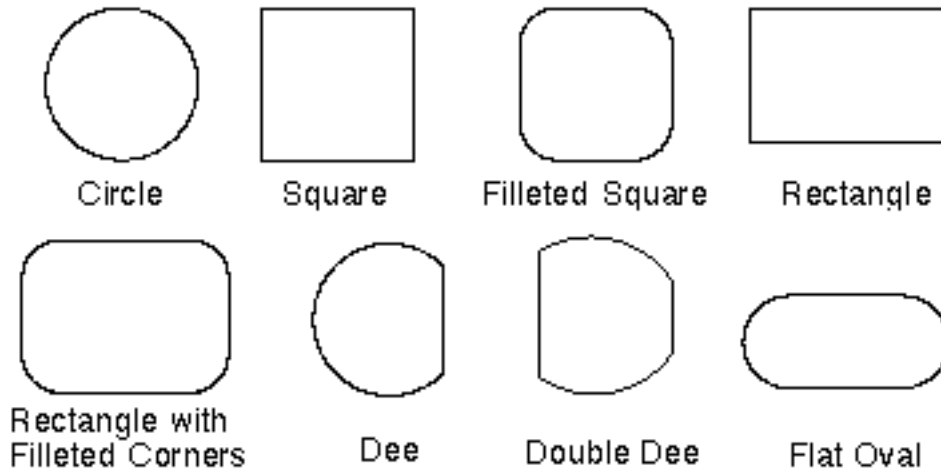
The configuration command (CONFIG) defines the configuration of the punching machine, including turret data. Indexable turret stations are supported.

Punch Definition

You can store definitions for several punches in a punch library. You can then activate this punch library in subsequent JCFs and use any of the library's punches, without having to redefine them.

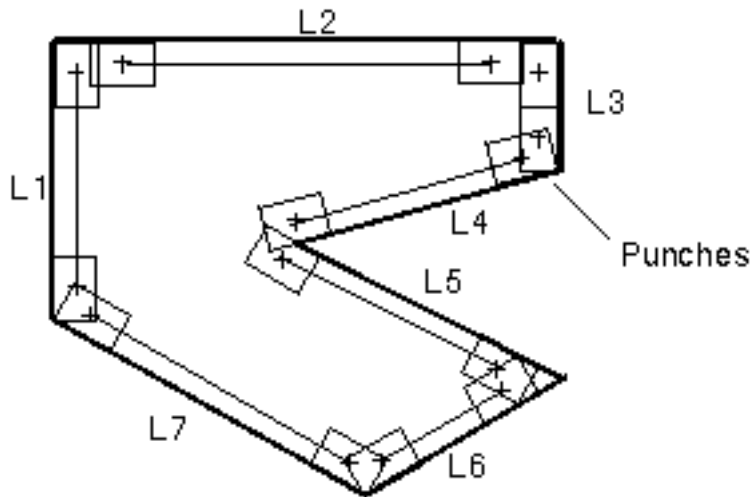
CVNC-P2 supports eight standard punch shapes: circle, square, square with filleted corners, rectangle, rectangle with filleted corners, flat oval, dee (circle with

single cutoff), and double dee (circle with two parallel cutoffs). You can also define nonstandard punch shapes.



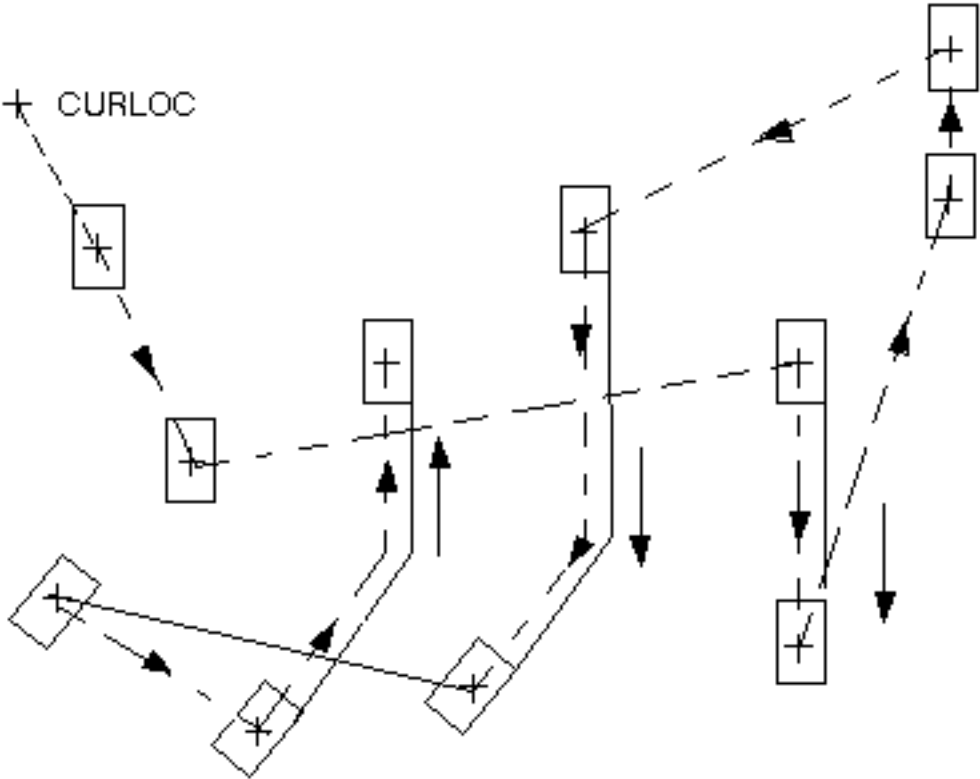
Variable Angle Nibbling and Punching

CVNC-P2 can punch or nibble multiple boundaries at different angles of orientation with the same punch.



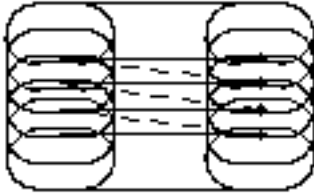
Sorting and Optimizing Punches

You can sort and optimize punch hits and sequences of punch hits by the nearest neighbor algorithm and turret considerations. You can block off segments of the JCF that will not be sorted. In the following figure, the arrows indicate the order of sorted punches and punch sequences.

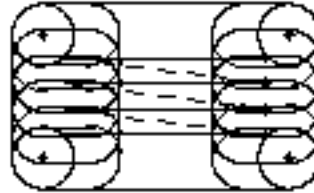


Area Clearance

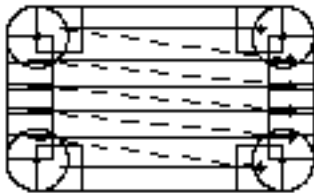
You can define an area-clearance (removal of stock in punch-size pieces) algorithm used for the following boundary shapes: oval, square, filleted square, rectangle, and filleted rectangle.



AREAOPT SINGLE



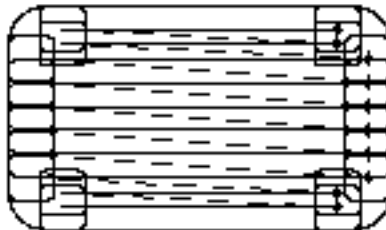
AREAOPT DOUBLE FILLPNCH



AREAOPT DOUBLE



AREAOPT NOCORNER



AREAOPT NOCORNER FILLPNCH

Configurable Output

CVNC-P2 supplies CVMAC macros with imbedded postprocessor commands (pass-through statements). You can modify these macros for site-specific output.

Automatic Search for Optimum Punch

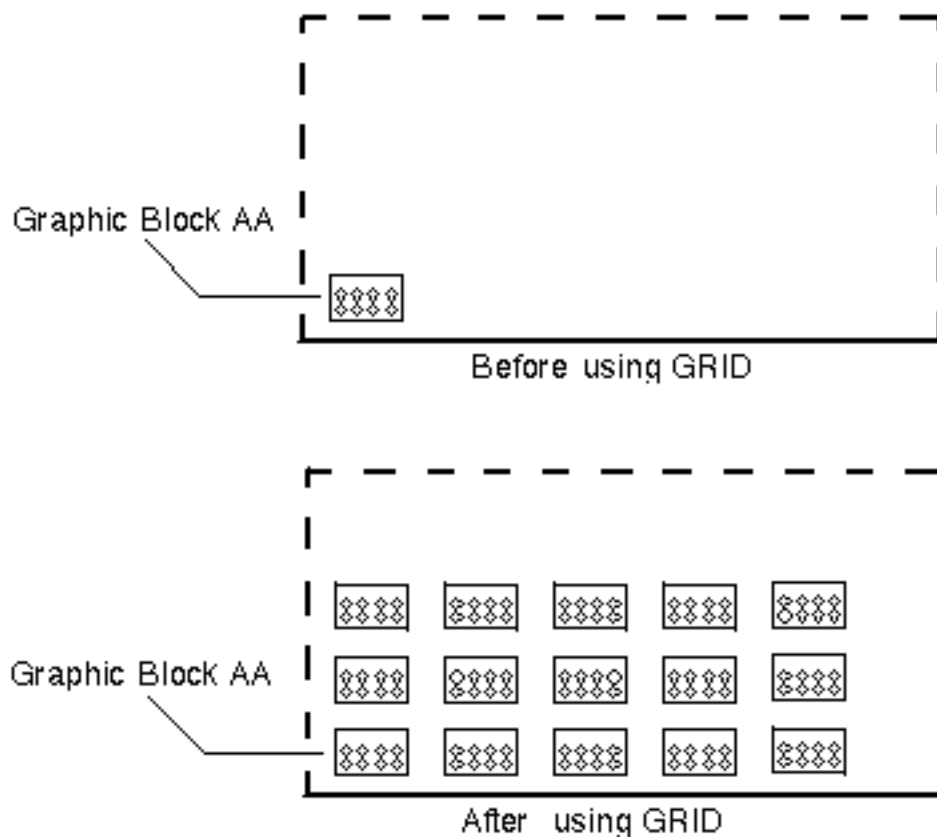
CVNC-P2 searches for the best punch to be used when punching and nibbling. You can use the SEARCH command to specify searching options, or search according to default options.

Repeating a Punching Operations Block

CVNC-P2 provides a macro (GRID) that repeats a block of punching operations a defined number of times along the x- and y-axes or within a maximum distance along the x- and y-axes.

GRID is used to repeat block AA five times along the x-axis and three times along the y-axis. The vertical distance between each graphic block is .2 and the horizontal distance is .25.

```
NC:> GRID 'AA' XNUM 5 YNUM 3 LOWLEFT d☞; UPRIGHT d☞; VERPUNWD .2  
HORPUNWD .25
```



For more information, see the *CVNC-P2 User Guide* and *CVNC-P2 Command Reference*.

CVNC Batch

You can use CVNC Batch for 2-axis, 3-axis and 5-axis milling, that is, in CVNC-M2, CVNC-M3 and CVNC-M5. You cannot use it for CVNC-T2 and CVNC-P2.

Creating a JCF

To get started you need to

- Create the JCF

To create a new JCF, use PREPARE along with the necessary command in CADD5 CVNC. You could be starting with either a new JCF or an existing JCF. You must use PREPARE to prepare an existing JCF for batch processing.

- When it opens, enter the relevant machining commands

Executing a JCF

During JCF execution, Batch CVNC does the following:

- Generates the complete CVNC file set. You may return this to CADD5 CVNC for tool path graphics generation and/or pass it to the separate output generator.
- Creates new CPLs, if required.
- Generates local output (if any) requested in the JCF.

To execute the JCF in Batch CVNC, use the following command at the UNIX prompt.

Enter

```
execnc <JCF file name>
```

where

<JCF file name> is a JCF name in UNIX filename format.

The Batch CVNC module executes and processes the JCF for the appropriate CVNC module, that is, M2, M3 or M5. The results of batch processing are the executed binary files and the associated tool path data.

Viewing Results

You can see the results of batch processing, that is, the toolpath graphics and errors in the VIEW mode. Use VIEW along with the necessary command in CADD5 CVNC, to enter this mode.

You can also use the default CADD5 CVNC mode to see the toolpath graphics.

This chapter describes each step in the NC solution, from creation of a CADDs model to machining. To help illustrate the process, the instructions in this chapter follow a sample part from design through production. The sample part used in this book is a mounting bracket that is used to fasten a piece of photographic equipment in the cargo-bay area of the space shuttle.

- Creating a CAM-Ready CADDs Model
- Designing the Mounting Bracket
- Creating the JCF
- Job Setup: Machine Configuration
- Job Setup: Defining Tools
- Job Setup: Generating Output
- Operation Setup
- Tool Motion Commands
- Creating the JCF for the Mounting Bracket
- Generating Output
- Generating Output for the Mounting Bracket
- Postprocessing
- Postprocessing Output for the Mounting Bracket
- Distributing Data

Creating a CAM-Ready CADDs Model

The first step in the NC solution is to create the CADDs model. The model is stored in the CADDs database, which is accessed by CVNC to create a part program. To create a CAM-ready CADDs model, several issues should be considered. To illustrate the steps in the CVNC solution, a sample part will be taken through the steps from design to production.

The sample part designers considered manufacturing needs when they created the part model so that it will not need modification for CVNC. The CAM-ready model was created with the following in mind:

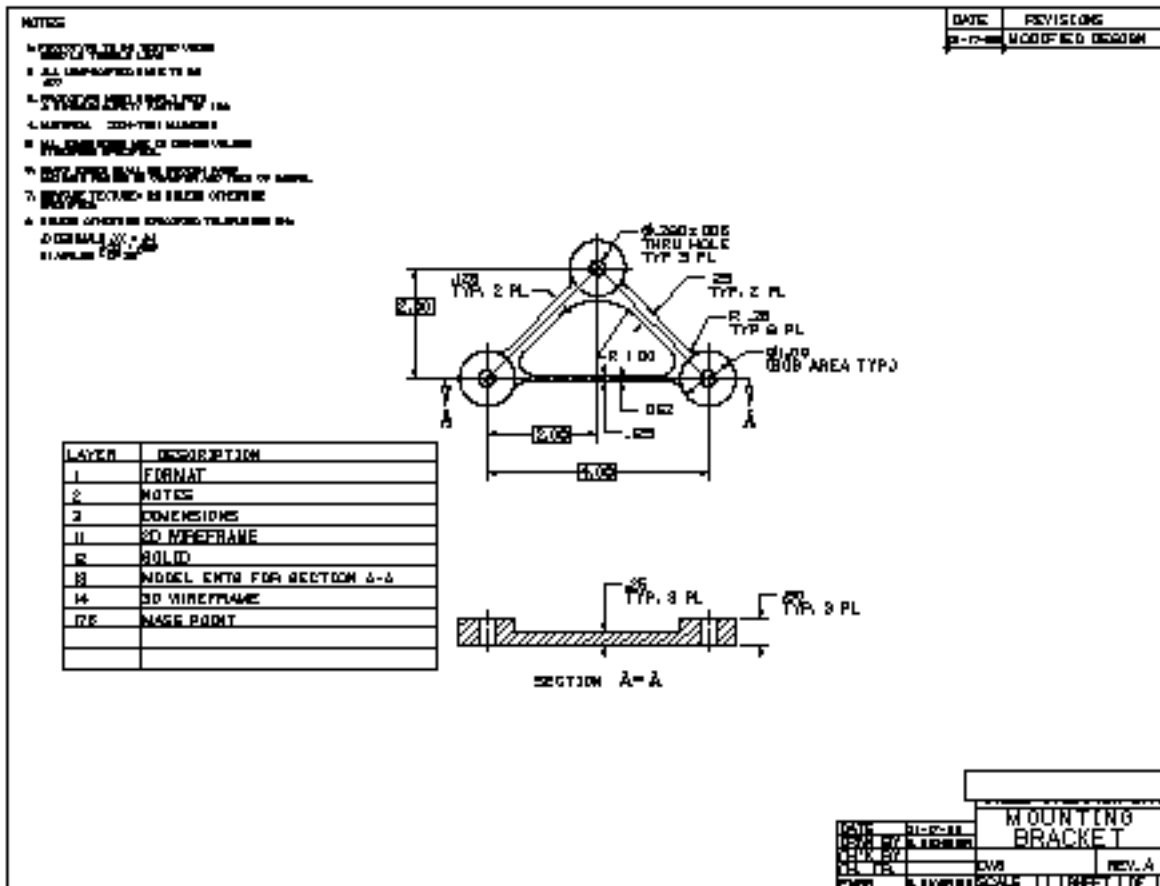
- Part geometry is true size. No dimensions are changed without also changing the geometry.
- Part geometry does not include excess or duplicate data.
- Entities are identified with unique tags.
- Entities are contiguous; there are no gaps between entities or surfaces.
- Fillets, break edges (small chamfers) are represented in the model geometry.
- Layering is used to separate data in an effective and standardized manner. For example, surface geometry is placed on separate layers from wireframe geometry-Plcaf3.
- The part illustrates the expected machining results. (Construction geometry can be used to accomplish this.)

Designing the Mounting Bracket

The mounting bracket CADDSS model was created with manufacturing needs and requirements in mind. Our sample part, the mounting bracket, is used to fasten a piece of photographic equipment in the cargo-bay area of the space shuttle.

The mounting bracket model was created using various CADDSS modeling products. The following applications were also used with the CADDSS model of the bracket:

- CVNC-M2 created a part program and generated a CLFile for machining the bracket.
- CVGPII postprocessed the CLFile.



Design and Manufacturing Worked Together

Design and manufacturing teams worked together throughout the mounting bracket production. Together, they agreed upon the following:

- A layering scheme must be established so that the part can be easily transferred among departments.
- Geometry should be functional for manufacturing purposes.
- The part coordinate system origin should be in a logical place for manufacturing purposes.
- Proper drawing views should be created with origins that relate to the machine coordinate system.

Layering Scheme

The design team began by establishing a standard layering convention, shown in the table below.

Table 3-1 Layering Scheme

| Layer | Contents |
|-----------|--|
| 0 | Information page; layering chart |
| 1 - 10 | Notes: dimensions, drawing entities, drawing formats, and title blocks |
| 11 - 75 | Model: lines, arcs, B-splines, Cpoles, solids (12) |
| 76 - 100 | Surfaces: Spoles, Nsurfs |
| 101 - 150 | NC tool paths (layers in this range will be selected when CVNC is generating tool paths) |
| 151 - 200 | Mass points, Finite element nodes and elements |
| 201 - 255 | Construction layers, temporary construction geometry |

Layering allows the part to be easily transferred among design, drafting, and manufacturing, because it separates data into more usable categories.

For example, layer 12 is reserved for solid modeling and surface shading. Manufacturing may not want to use shaded surfaces when generating tool paths because these surfaces may be too small and complex to be used efficiently. The manufacturing engineer can simply exclude layer 12 when echoing layers.

A color workstation can discriminate layer ranges by color, which further heightens the distinctions made and groupings defined by the layering scheme.

Shaded surfaces may be useful after the JCFs have been written. Part programmers can run JCFs on the solid images to better visualize step-by-step material removal.

Functional Geometry

The geometry of the part must be valid for CVNC. When designing the mounting bracket, the design engineers were sure to

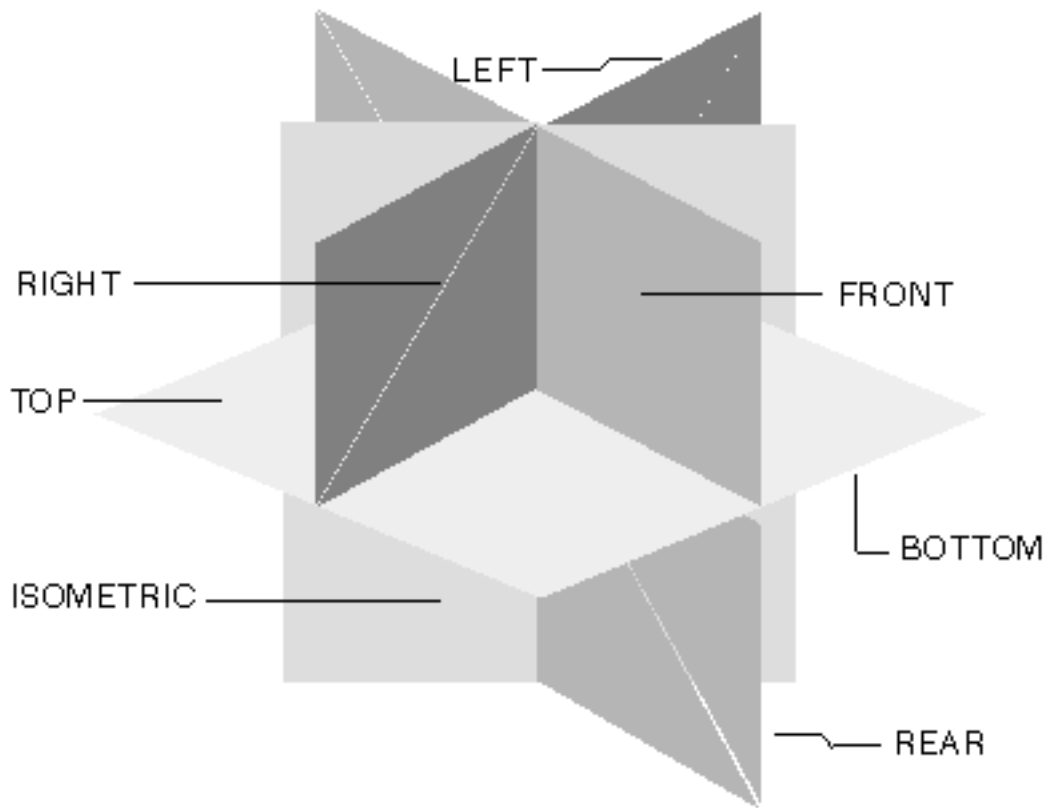
- Give every entity a unique tag
- Make sure that entities were contiguous
- Provide a detailed drawing with the necessary notes and dimensions for the geometry

A common manufacturing problem is that a part design often includes too much geometry for manufacturing. As discussed earlier, layering is a way of separating geometry so as to provide only what is needed.

Views

Views allow the design or manufacturing engineer to select ways of seeing model geometry. CADDs defines views with respect to construction planes (Cplanes).

There are seven standard Cplanes: TOP, RIGHT, LEFT, FRONT, BOTTOM, REAR, and ISOMETRIC, as shown in the following figure.



The detailed drawing of the mounting bracket shown at the beginning of this section is defined at the TOP construction plane.

Another drawing of the mounting bracket (which you will see in “Creating the JCF” on page 3-8” contains four views:

- TOP
- FRONT
- BACK
- ISOMETRIC

The front view, for example, was necessary when examining drill tool paths. With a top view, you cannot inspect the depth to which the tool is drilling.

Define a view to allow visualization of the machine coordinate system. The machine coordinate system and machine zero are defined with the CVNC

DATUM command. The view origin is defined in CADD5 with the DEFINE VIEW CPL command.

Creating the JCF

CVNC allows you to create, execute, and store NC part programs. CVNC commands are stored in a Job Control File (JCF), which is a part program containing references to tagged entities of the CADDSS model. It is an ASCII text file that can be reexecuted and modified.

How Is the JCF Created

As you enter CVNC commands, the JCF is automatically generated. The JCF contains a permanent part program record. You can access it and reexecute or edit the part program as needed.

Each command is processed immediately, providing tool path graphics, as appropriate. As you execute each command, you can examine the resulting tool motion and parameter changes (contained in system variables). Alternatively, you can batch process the JCF using CVNC Batch.

You can change any line using editing commands that allow you to step backward, modify the JCF, and step forward. As you edit the JCF, the associated graphics update.

Writing a JCF consists of entering a command sequence to define machining conditions for one or more cutting tools and to create tool paths. Typically, one JCF corresponds to a single setup on a machine tool.

JCF Contents

The JCF includes

- System commands
- Application commands

A JCF can also include pass-through statements (statements that pass directly to the output file and do not affect CVNC processing), and commands that call CVMAC macros.

You can use CVNC editor commands to view and modify the JCF. These commands are not recorded in the JCF.

System Commands in the JCF

System commands let you

- EXIT, FILE, and QUIT your JCF
- Change the name of your JCF, and save it during the work session
- Control system graphics
- Resume to CADDs level
- Monitor and manipulate system variables
- Generate, call, and access command files
- Add or modify your CADDs part geometry while you are in the CVNC environment

Application Commands in the JCF

Application commands are often application-specific. NC commands allow you to determine

- Job setup
- Operation setup
- Tool Motion

Application commands are discussed in more detail in the following sections.

Job Setup: Machine Configuration

Job setup commands specify the parameters required for the entire NC process. These parameters include machine configuration, tool definition, and output format.

Configuring the Machine

All CVNC products provide a CONFIG command for configuring the machine tool, as noted in the three examples below.

The following CVNC-M2 CONFIG command configures a machine with a BAXIS rotary table (rotation about the y-axis) and an AAXIS rotary head (see the figure, Machine with AAXIS Rotary Head and BAXIS Rotary Table, on the next page). Note that A, B, and C are used to denote axes of rotation.

```
NC:> CONFIG AAXIS HEAD X -2 Y 4.5 Z 5; BAXIS X -4 Y 6.5 Z 5;
```

You can also specify the range of motion of each axis of the target machine tool (see the following command.) All requested motions are then checked with these limits.

```
NC:> CONFIG XAXIS MAXIMUM exp MINIMUM exp
```

The following CVNC-P2 CONFIG commands describe a punching machine and its turret configuration:

```
NC:> CONFIG MACHINE '1' CYCLE MAXTUR 4 MAXTON 100 INCH  
NC:> CONFIG ROT B CAXIS  
NC:> CONFIG SIZE C LOWRLIM 1.5 UPPRLIM 3  
NC:> CONFIG STATION 1 SIZE C ROT B
```

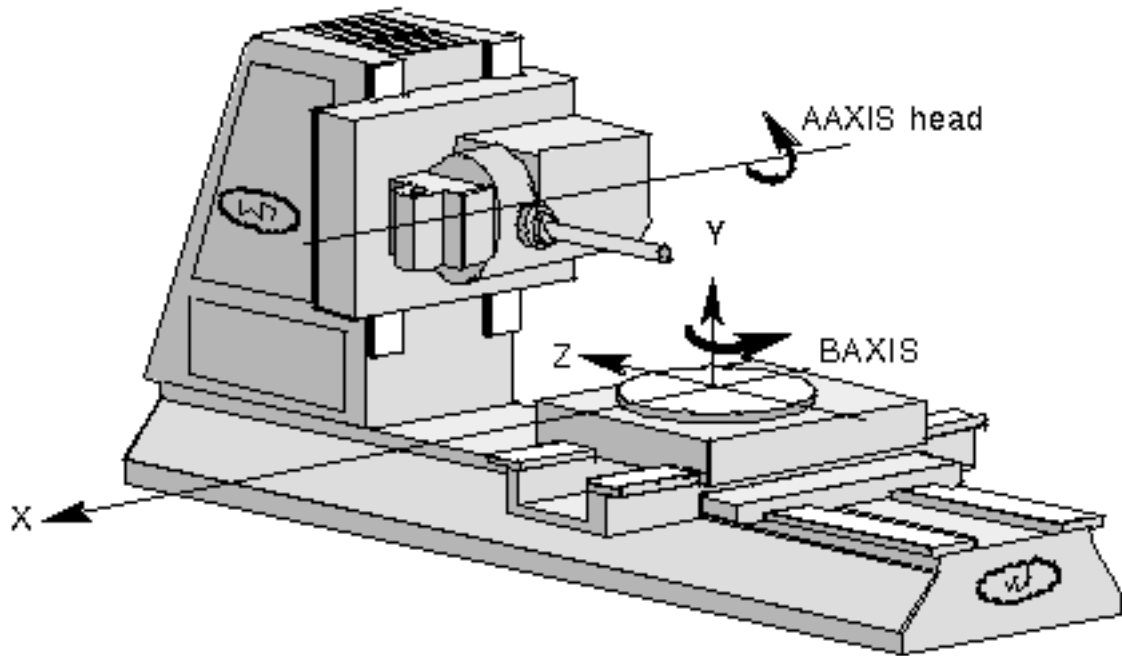
The punch press supports cycle statements, has four turret stations, and has a maximum tonnage of 100 tons.

Rotation class B is defined so that a punch can be inserted with CAXIS capability (indexable turret). Size C is defined to accept punches within the range of 1.5-3 inches. Station 1 is assigned size class C and rotation class B.

The following CVNC-T2 CONFIG command defines a machine with the primary turret in the rear quadrant of the machine configuration relative to spindle center line. The x-differential between the two turrets on a twin-turret lathe is 4.0. The y-differential between the two turrets on a twin-turret lathe is -12.0.

```
NC:> CONFIG REAR XDIF 4.0 YDIF -12.0
```


Figure 3-1 Machine with AAXIS Rotary Head and BAXIS Rotary Table



Job Setup: Defining Tools

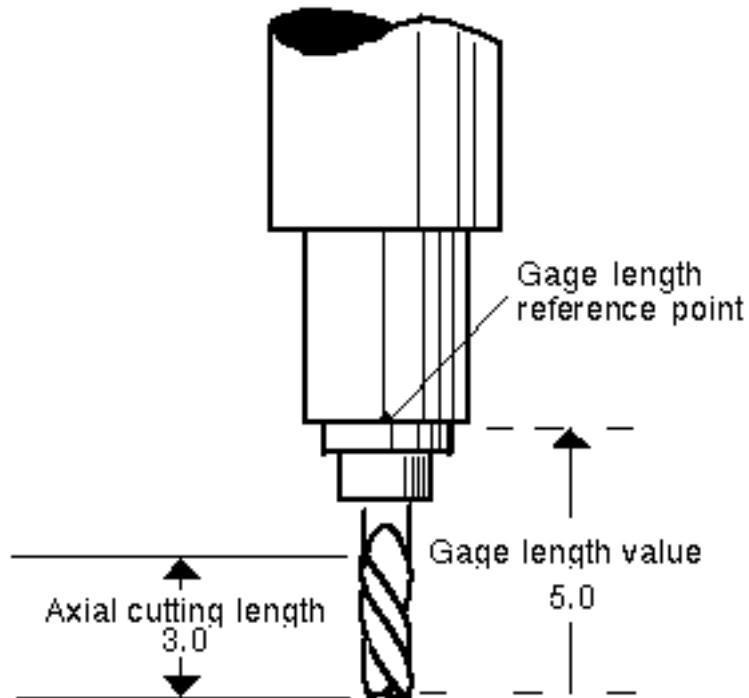
Job setup commands specify the parameters required for the entire NC process. These parameters include machine configuration, tool definition, and output format.

Defining Tools

Milling and turning are defined with the DEFTOOL command. Punches are defined with a DEFPUNCH command.

The following DEFTOOL command defines a mill with a .25-inch diameter, a 3-inch height, and a 5-inch gage and assigns the name 'MILL5.0'.

```
NC:> DEFTOOL 'MILL5.0' MILL7 DIA .25 CORNER 0 HEIGHT 3.0 GAGE 5.0
```



Job Setup: Generating Output

Job setup commands perform tasks such as specifying output.

Specifying Output Format

The OUTPUT command specifies the output format (CLFile, APT, or COMPACT II) and file name. OUTPUT can also specify that only portions of a JCF are output.

The following OUTPUT command specifies that the output file is named TOP-DRILL and is in CLFile format.

NC:> OUTPUT CLFILE TEXT 'TOP-DRILL'

JCF

```
[JCF  CVNC  MILL      ]
AUDIT  ON
PARTNO  'BRACKET'
MACHIN  'MILL'  1
SELECT  LAY  120
OUTPUT  CLFILE  TEXT  'TOP-DRILL'
PLANE   ZREF  0
PLANE   ZAPPR      1.04  ZREF
PLANE   ZCLEAR    1      ZREF
PLANE   ZRETRACT  .05   ZREF
PLANE   ZWORK     -.5   ZREF
DEFTOOL 'SPOT-2'  DRILL  DIA  .1  GAGE  2.0
DEFTOOL 'DR-280'  DRILL  DIA  .280 GAGE  2.0
DATUM   'TOP'
HOMEPT  FROM  X  -3  Y  1  Z  5;
```

TEXT modifier creates a text version of the CLFile, in addition to the binary files.

Output File

```
PPRINT  PARTNO  'BRACKET'
PARTNO   BRACKET
PPRINT  MACHIN  'MILL'  1
MACHIN   MILL  1.0000
PPRINT  SELECT  LAY  120
PPRINT  OUTPUT  CLFILE  TEXT  'TOP-DRILL'
PPRINT  PLANE   ZREF  0
PPRINT  PLANE   ZAPPR      1.04  ZREF
PPRINT  PLANE   ZCLEAR    1      ZREF
PPRINT  PLANE   ZRETRACT  .05   ZREF
PPRINT  PLANE   ZWORK     -.5   ZREF
PPRINT  DEFTOOL 'SPOT-2'  DRILL  DIA  .1  GAGE  2.0
PPRINT  DEFTOOL 'DR-280'  DRILL  DIA  .280 GAGE  2.0
PPRINT  DATUM   'TOP'
PPRINT  HOMEPT  FROM  X  -3  Y  1  Z  5;
FROM    -3.0000,    1.0000,    5.0000
```

TEXT modifier creates a text version of the CLFile, in addition to the binary files.

Operation Setup

Operation setup commands specify parameters that regulate individual cutting operations, such as

- Tool selection
- Active construction plane (CPL) selection
- Entity grouping
- Feeds, speeds, z-planes, and tolerance

There are also application-specific operation setup commands, such as the CVNC-P2 command that defines the method of sorting punch hits (DEFSORT).

Selecting Tools - CHGTOOL

All CVNC products use the CHGTOOL command to select a tool.

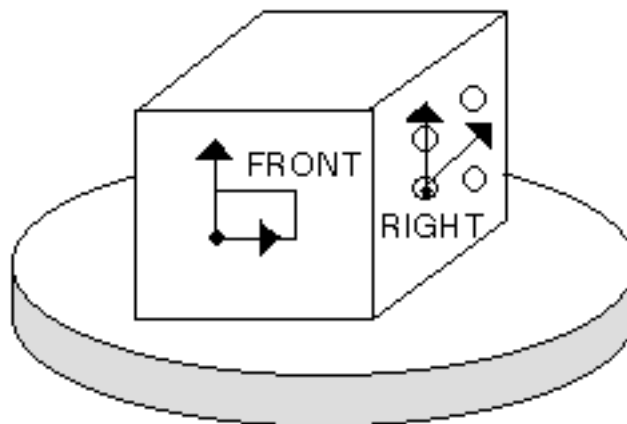
You can select tools that you defined in the current JCF or tools that were defined in a previous JCF and are in the active tool library.

Selecting the Active Construction Plane - INDEX

CVNC allows you to specify the active construction plane. In CVNC-M2/M3/M5, you can index to create a new construction plane.

The following CVNC-M2 command indexes to the RIGHT construction plane:

```
NC:> INDEX "RIGHT"
```



Grouping Entities - NCGROUP

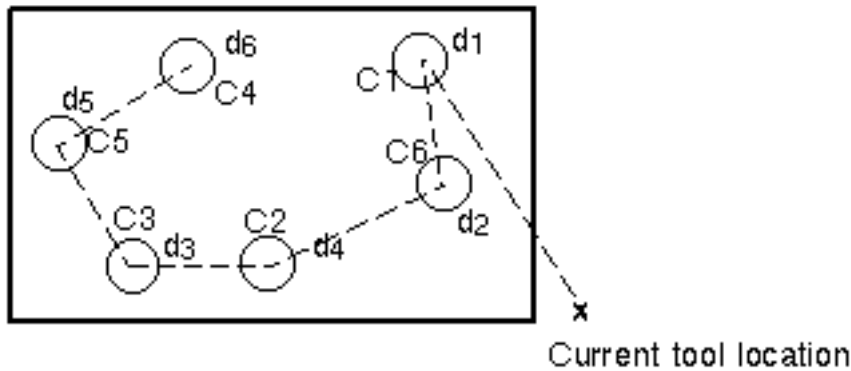
As described in Chapter 2, “CVNC Features”, you can group entities into named groups. This is useful when using the same entities or locations in multiple operations, such as

- Pocketing/profiling
- Drilling/countersinking/boring/tapping

To verify selections, you can highlight them in a different color or font.

The following command groups together six holes. It also specifies optimization, which changes their order by applying a nearest-neighbor algorithm, starting with the first entity listed in the NCGROUP.

```
NC:> NCGROUP AA OPTIM ENT: MODEL ent d1 d2 d3 d4 d5 d6; NCGROUP AA  
OPTIM $C1 $C6 $C3 $C2 $C5 $C4;  
NC:> DRILL DEPTH .3 ENT AA;
```

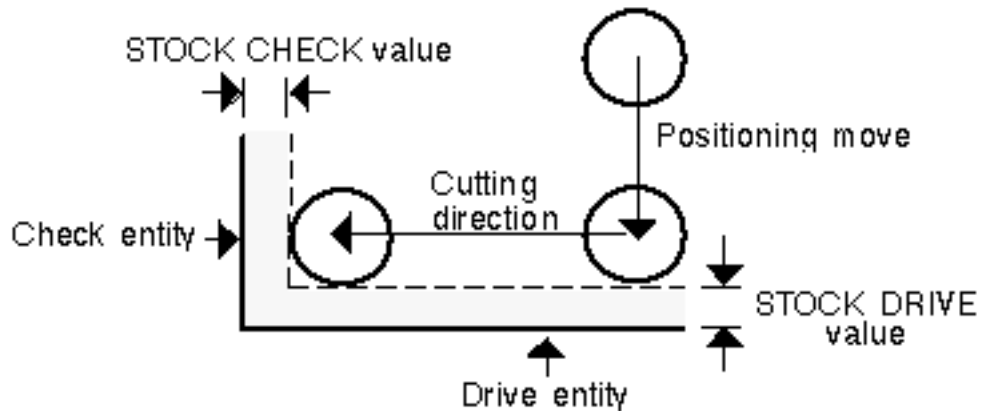


Defining Operation Parameters

You can define operation parameters, such as

- Feed rate - FEED
- Spindle speed - SPEED
- Stock offsets (thicks) - STOCK
- Tolerance - TOLER
- Setting z-planes - PLANE

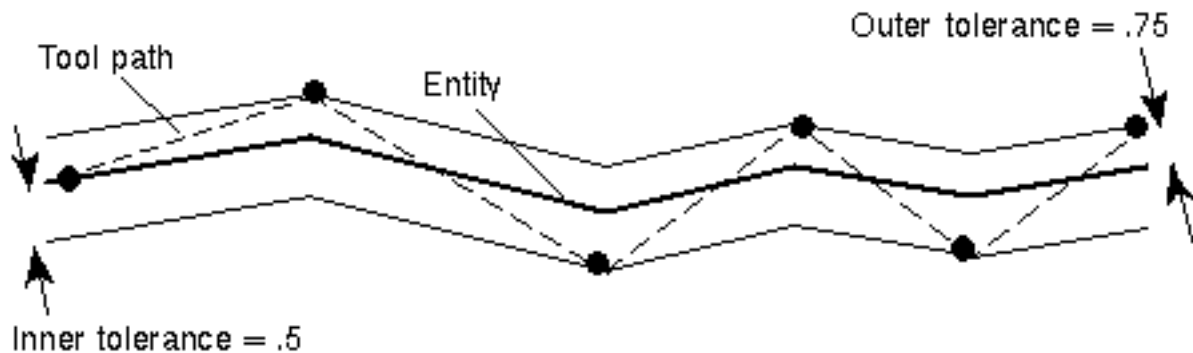
The CVNC-M2 STOCK command can assign stock to drive and check entities, and it can assign individual stock values to separate entities.



The TOLER command specifies a tolerance zone applied to each tool path side when cutting circles, arcs, arc segments of B-splines, conics, or Cpoles.

INTOL specifies the maximum amount of material that may be left on the part; OUTTOL specifies the maximum amount of material that may be removed. The tool path will stay within the INTOL and OUTTOL tolerance values, as in the following example:

```
NC:> TOLER INTOL .5 OUTTOL .75
```

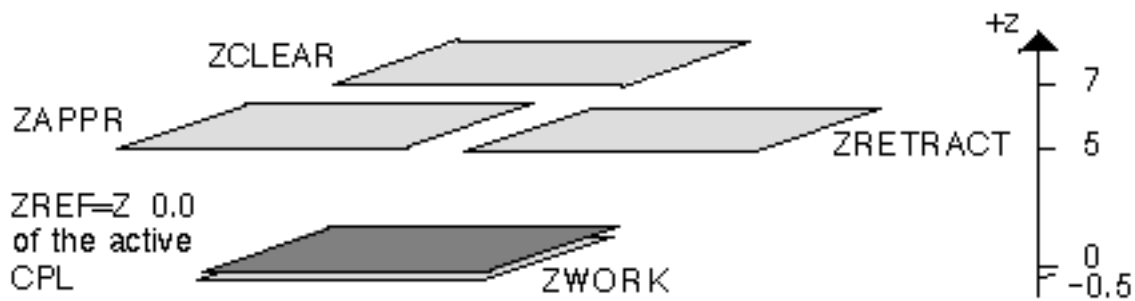


The PLANE command sets up a z-plane that can be referenced during cutting and noncutting movements.

The following commands establish the four z-plane types:

```
NC:> PLANE ZREF 0
NC:> PLANE ZAPPR 5 ZREF
NC:> PLANE ZWORK -.5 ZREF
NC:> PLANE ZRETRACT 5 ZREF
```

```
NC:> PLANE ZCLEAR 7 ZREF
```



Application-specific Parameters

Each CVNC application has operation setup commands specific to its operations.
For example:

- With the CVNC-M2 CALCRAD command, you can calculate the tool offset necessary to cut an entity when the entity height is at the same level as the tool corner radius.
- With the CVNC-P2 DEFSORT command, you can reorder output punch hits and cycles.

Tool Motion Commands

CVNC provides cutting and noncutting movement. Examples of noncutting motion commands are

- **MOVE** - Moves the cutter from its current location to a new location at the **RAPID** feed rate (feed rates are defined with another command, **FEED**).
- **RETRACT** - Moves the cutter from its current location to a new location at the **RETRACT** feed rate.

Cutting Motion for CVNC-M2

Cutting motion for 2- and 2 1/2-axis milling includes

- Hole processing (drilling, boring, countersinking, and tapping)
- Incremental and absolute cutting motion
- Cutting motion along individual entities
- Profiling
- Pocketing (lace or unidirectional)
- Face milling

For more information on CVNC-M2 tool motion, see the CVNC-M2 features listed in Chapter 2.

Cutting Motion for CVNC-M3

Cutting motion for 3-axis milling includes

- Surface machining (straight line, parametric, and control curve)
- Surface intersection machining
- Projecting 2- and 2 1/2-axis motion onto three-dimensional surfaces
- 3-dimensional contouring
- Waterline machining and access hole processing operations on the roughing material model

For more information on CVNC-M3 tool motion, see the CVNC-M3 features listed in Chapter 2, “CVNC Features”.

Cutting Motion for CVNC-M5

Cutting motion for 5-axis milling includes

- Surface machining (straight line, parametric and control curve)
- Profiling
- Surface intersection machining
- Swarfcutting

Cutting motion for 4-axis milling includes parametric surface machining and 4-axis surface intersection machining.

For more information on CVNC-M5 tool motion, see the CVNC-M5 features listed in Chapter 2, “CVNC Features”.

Cutting Motion for CVNC-T2

Cutting motion for 2-axis turning includes

- Continuous path profiling
- Rough turning/boring/facing at any angle

For more information on CVNC-T2 tool motion, see the CVNC-T2 features listed in Chapter 2, “CVNC Features”.

Cutting Motion for CVNC-P2

Cutting motion for 2-axis punching includes

- Punching
- Nibbling

For more information on CVNC-P2 tool motion, see the CVNC-P2 features listed in Chapter 2, “CVNC Features”.

Creating the JCF for the Mounting Bracket

JCFs were created for the different operations involved in machining the mounting bracket:

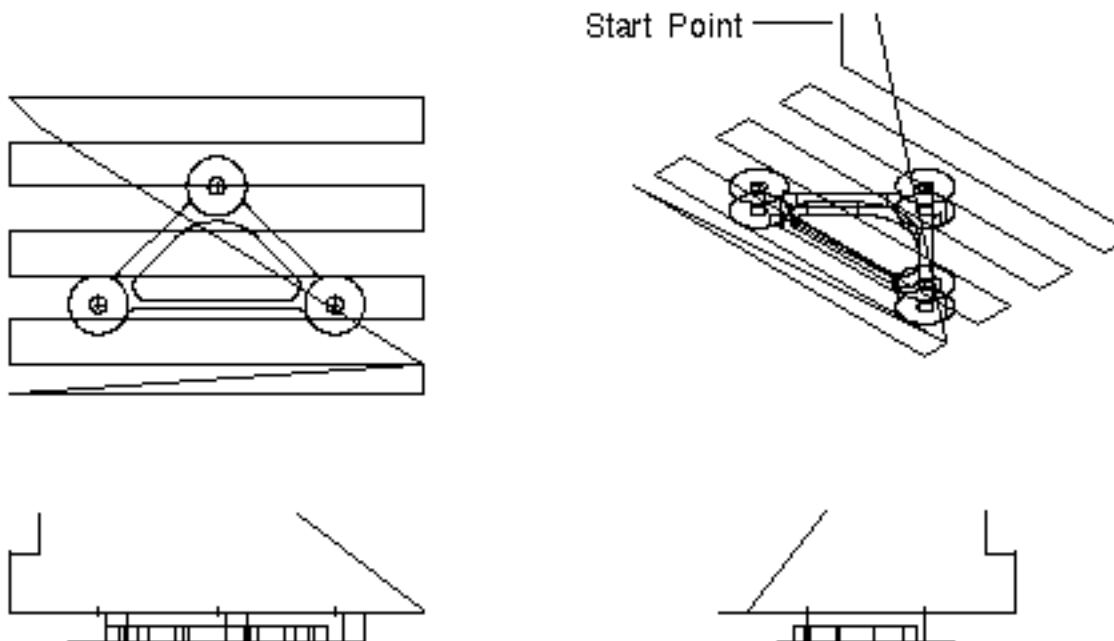
- OPR 1 clears away the bottom of the stock using the AREAMILL lace pocketing command.
- OPR 2 drills the three holes in the bracket.
- OPR 3 uses AREAMILL, CUT, and PROFILE to finish machining the bracket.

All three JCFs call a command file named PLANES, which contains PLANE commands used to establish reference planes. By storing these commands in a command file, the programmer need not enter duplicate commands for each JCF, but can just call the command file.

Operation 1 (JCF OPR 1)

The first operation machines the stock flat with the AREAMILL command, cutting off .010 of the material, as shown below in Figure 3-2, “Area Clearance,” on page 3-21. AREAMILL clears an area by lace pocketing. In the JCF on the next page, we decomposed the CALL command (using the IN command), to see the contents of the command file, PLANES.

Figure 3-2 Area Clearance



The following figure shows the entire JCF for operation 1.

Figure 3-3 JCF Operation 1

```

[JCF  CVNC  MILL  ]
AUDIT  OF
#>*
#>*  SIDE 1  ..  OPERATION  1
#>*
#>*  RAW STOCK IS ALUMINIUM, APPROX SIZE IS
#>*  MILL STOCK FLAT, APPROXIMATELY A .010" CUT.
#>*
#>*  WHEN THIS OPERATION IS COMPLETE TURN PAET OVER.
#>*
OUIFUI  CLFILE  TEXT  'BOTTOM-MILL'
SELECT  LAY  105
#>*  ZPLANE TOP = 0
READ  'TOP'  TOPPLANE
COMLIB  'SETUP'
[CALL  PLANES]
>[CMFIL  PLANES]
PLANE  ZREF  0
PLANE  ZAPPR  1.04  ZREF
PLANE  ZCLEAR  1  ZREF
PLANE  ZRETRACT  .05  ZREF
PLANE  ZWORK  -.5  ZREF
[EOF ]

DEFTOCL  'MILL2-1'  MILL7  DIA  1  CORNER  0  HEIGHT  3.0
HOMEPT  FROM  X  -3  Y  1  Z  5;
CLEAR  HOME
CHSTOCL  3  'MILL2-1'
SPEED  RPM  1800  CLW
FEED  APPROACH  IPM  80
CALL  PLANES
FEED  CLEAR  RAPID
FEED  CONNECT  IPM  20
FEED  CUT  IPM  20
FEED  PLUNCE  IPM  12
FEED  RETRACT  IPM  40
COOLANT  MIST
PLANE  ZWORK  0  ZREF
AREAMILL  LACE  OUTDIR  NANIS  BOUND  LOC  X  -4  Y  -4  X  4  Y  -4  X  4  Y
2  X  -4  Y<*>  2;
RETRACT
COOLANT  OFF
CLEAR  HOME
[DONE]
    
```

The READ command allows for user input. In this JCF, 0 was input.

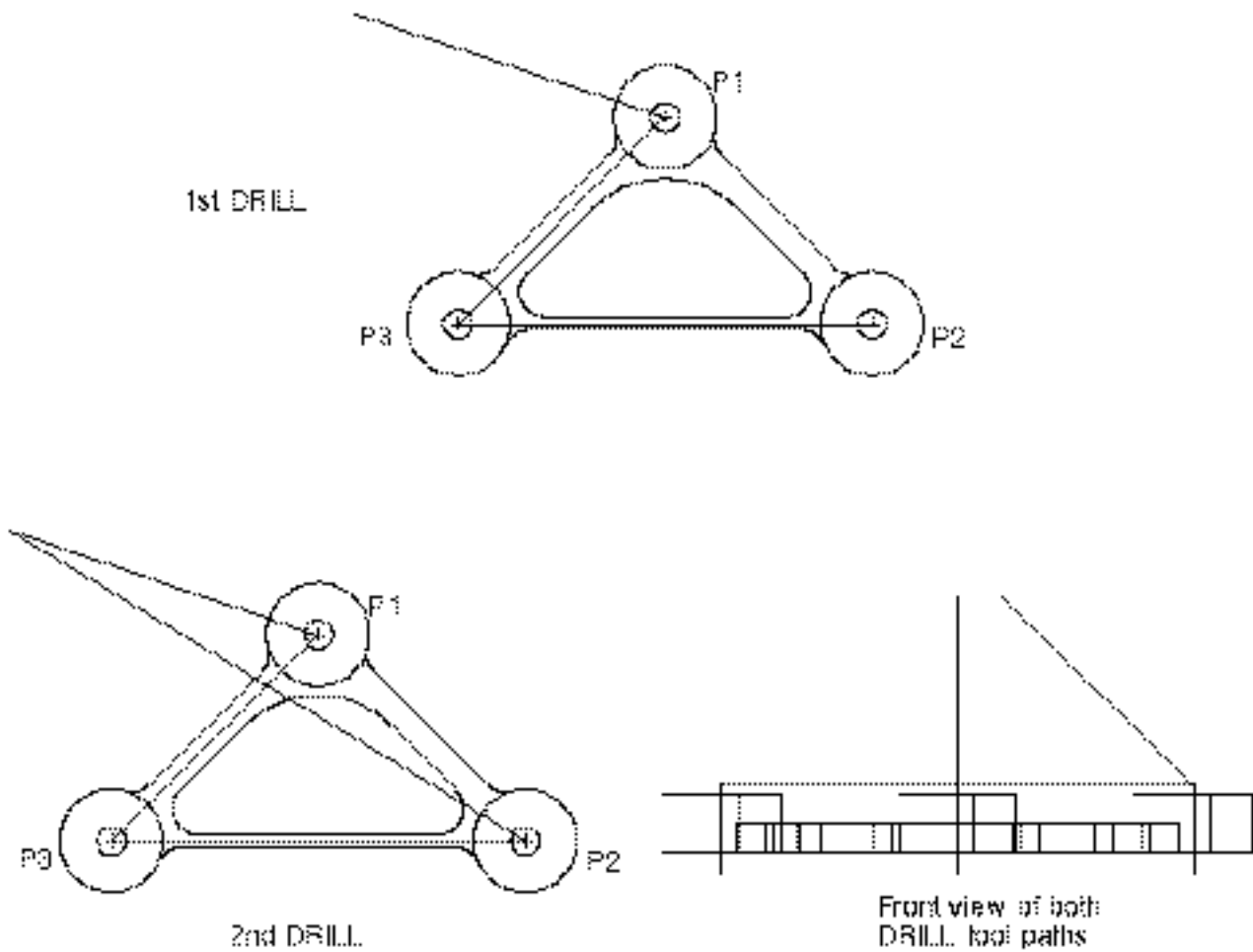
Contents of command file, PLANES

Operation 2 (JCF OPR 2)

Before running the second operation, the part must be turned over. An NCGROUP was formed to group the three hole entities to be drilled. OPR 2 drills the NCGROUP with two drilling passes, as shown in Figure 3-4, “Drilling,” on page 3-24.

- The first drill is a spot drill. It uses a .1 diameter drill and drills .05 inch.
- The second drill uses a .280 diameter drill and drills .6 inch.

Figure 3-4 Drilling



The following figure shows the entire JCF for operation 2.

Figure 3-5 JCF Operation 2

```
[JCF  CVNC  MILL ]
AUDIT ON
#>*
#>*  SIDE 2 - OPERATION 2
#>*  TURN STOCK OVER.
#>*  FIXTURE RAW STOCK ON A FLAT SUBPLATE BY DOUBLE-SIDED
TAPE.
#>*  SPOT DRILL AND DRILL 3 .280 HOLES THRU
#>*
OUTPUT CLFILE TEXT  'TOP-DRILL'
SELECT LAY 110
#>*  ZPLANE TOP = 0
READ 'TOP'  TOPPLANE _____ The READ command allows for
COMLIB 'SETUP' user input. In this JCF, 0 was input.
CALL PLANES _____ Call to command file, PLANES
DEFTOOL 'SPOT-2' DRILL DIA .1
DEFTOOL 'DR-280' DRILL DIA .280
HOMEPT FROM X -3 Y 1 Z 5;
CLEAR HOME
CHGTOOL 1 'SPOT-2'
SPEED RPM 3600 CLW
FEED APPROACH IPM 80
FEED CLEAR RAPID
FEED PLUNGE IPM 12
FEED RETRACT IPM 40
COOLANT MIST
NCGROUP HOLESET ENT $P1 $P2 $P3;
DRILL DEPTH .05 ENT HOLESET;
RETRACT
COOLANT OFF
CLEAR HOME
```

```
CHGTOOL 2 'DR-280'  
SPEED RPM 3600 CLW  
FEED APPROACH IPM 80  
FEED CLEAR RAPID  
FEED PLUNGE IPM 6  
FEED RETRACT IPM 40  
COOLANT MIST  
DRILL DEPTH .6 PECK MAXDEP .2 ENT HOLESET;  
RETRACT  
COOLANT OFF  
CLEAR HOME  
[DONE]
```

Operation 3 (JCF OPR 3)

The third operation finishes machining with three AREAMILL commands, several CUT commands, and two PROFILEs, as illustrated on the following pages.

- The first AREAMILL clears the top area of the stock (see Figure 3-6, “AREAMILL Clearing Top Area of Stock,” on page 3-27).
- The second and third AREAMILL machines the stock at different z-levels. The boss areas are specified as islands; face cutting is performed around them (see Figure 3-7, “AREAMILLS Around Islands,” on page 3-28).
- Several cuts are made around the bracket outer boundary (see Figure 3-8, “Cuts Around Outer Boundary of Bracket,” on page 3-28).
- Two PROFILEs are made on the inner boundary of the bracket at different z-levels (see Figure 3-9, “PROFILEs on Inner Boundary of Bracket,” on page 3-29).

The entire JCF for operation 3 is illustrated last in Figure 3-10, “JCF for Operation 3,” on page 3-30.

Figure 3-6 AREAMILL Clearing Top Area of Stock

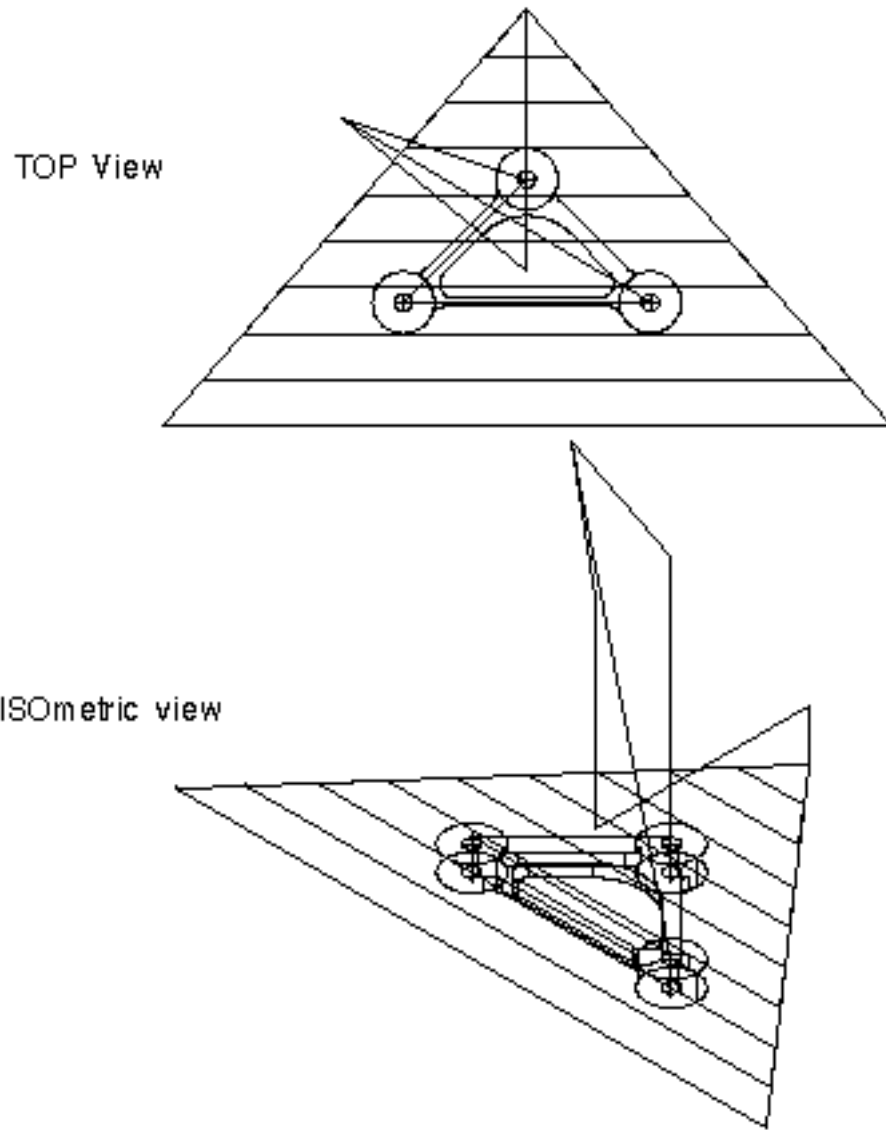


Figure 3-7 AREAMILLS Around Islands

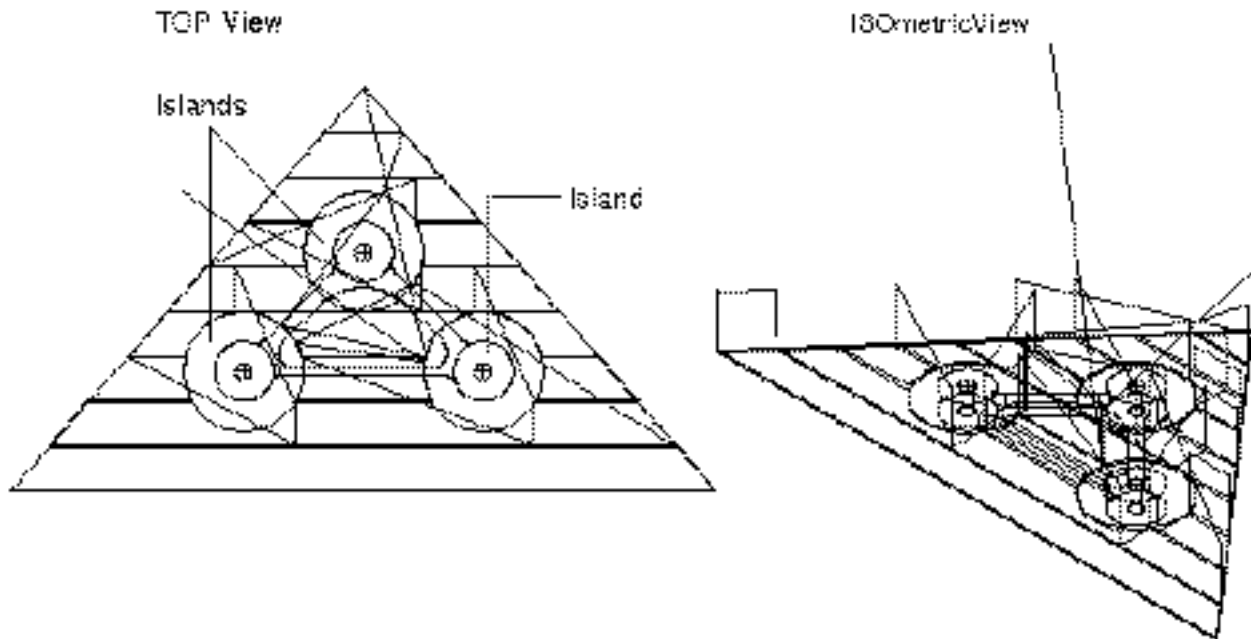


Figure 3-8 Cuts Around Outer Boundary of Bracket

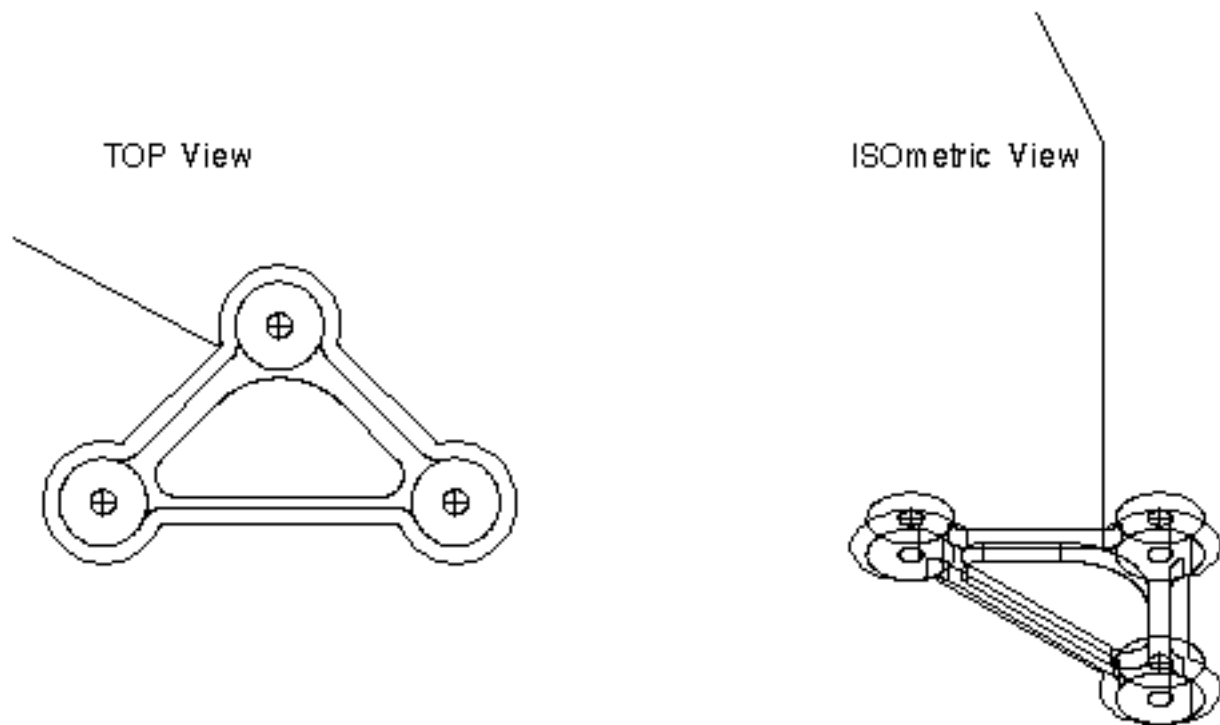


Figure 3-9 PROFILES on Inner Boundary of Bracket

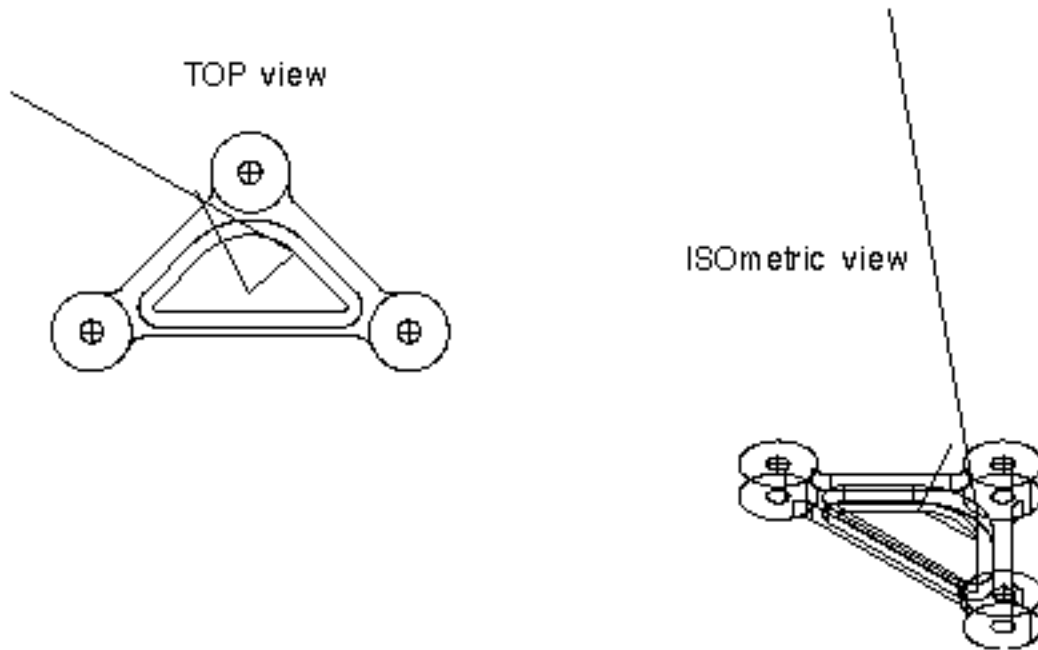


Figure 3-10 JCF for Operation 3

```
[JCF  CVNC  MILL  ]
AUDIT  ON
#>*
#>*  SIDE 2  -  OPERATION 3
#>*
OUTPUT  CLFILE  TEXT  'TOP-MILL'
SELECT  LAY  120
#>*  ZPLANE  TOP  =  0
READ  'TOP'  TOPPLANE  ← The READ command allows for
COMLIB  'SETUP'          user input. In this JCF, 0 was input.
CALL  PLANES
DELTOOL  'MILL2-1'
DEFTOOL  'MILL2-1'  MILL7  DIA  1  CORNER  0  HEIGHT  3.0
DELTOOL  'MILL2-2'
DEFTOOL  'MILL2-2'  MILLMILL7  DIA  .375  CORNER  0  HEIGHT  2.0
HOMEPT  FROM  X  -3  Y  1  Z  5;
CLEAR  HOME
CHGTOOL  3  'MILL2-1'
SPEED  RPM  1800  CLW
```

```

FEED APPROACH IPM 80
FEED CLEAR RAPID
FEED CONNECT IPM 6
FEED CUT IPM 8
FEED PLUNGE IPM 12
FEED RETRACT IPM 40
COOLANT MIST
MOVE NYLOC X 0 Y -1.5;
PLANE ZWORK 0 ZREF
AREAMILL LACE CUTDIR XANIS BOUND LOC X -7 Y -4.5 X 7 Y -4.5 X 0 Y
3.5;
RETRACT
SELECT LAY 125
PLANE ZWORK -.2 ZREF
AREAMILL LACE CUTDIR XANIS BOUND LOC X -7 Y -4.5 X 7 Y -4.5 X 0 Y
3.5;< *
ISLAND $C4;$C5;$C6;
RETRACT
PLANE ZWORK -.25 ZREF
AREAMILL LACE CUTDIR XANIS BOUND LOC X -7 Y -4.5 X 7 Y -4.5 X 0 Y
3.5;< *
ISLAND $C4;$C5;$C6;
RETRACT
CLEAR HOME
CHGTOOL 4 'MILL2-2'
SPEED RPM 3600 CLW
SELECT LAY 130
MOVE TO $L3;INTOF TO $A8;BIAS X -.6 Y -.3;
PLANE ZWORK -.375 ZREF
CUT TO $L3;INTOF TO $A8;BIAS X -.6 Y -.3;
CUT CHECK TO $A8;BIAS X -.6 Y -.3;TO $C4;CCLW BIAS X -.55 Y -.2;TO
$A5< *
;CLW BIAS X .55 Y -.2;
CUT CHECK TO $L1;CCLW BIAS X .6 Y -.3;
CUT CHECK TO $A4;BIAS X 1.65 Y -1.45;TO $C6;CCLW BIAS X
$A9;CLW BIAS X -1.8 Y -1.5;
CUT CHECK TO $L3;CCLW BIAS X -.6 Y -.3;
CUT CHECK TO $A8;BIAS X -.6 Y -.3;
PLANE ZWORK -.505 ZREF
    
```

```
CUT TO $L3;INTOF TO $A8;BIAS X -.6 Y -.3;
CUT CHECK TO $A8;BIAS X -.6 Y -.3;TO $C4;CCLW BIAS X -.55 Y
-.2;TO $A5<*
;CLW BIAS X .55 Y -.2;
CUT CHECK TO $L1;CCLW BIAS X .6 Y -.3;
CUT CHECK TO $A4;BIAS X 1.65 Y -1.45;TO $C6;CCLW BIAS X 1.8 Y
-1.5;TO <*
$A7;CLW BIAS X 1.5 Y -2.2;
CUT CHECK TO $L2;CCLW BIAS X 1.3 Y -2.1;
CUT CHECK TO $A6;BIAS X -1.3 Y -2.1;TO $C5;CCLW BIAS X
-1.5 Y -2.2;TO <*
$A9;CLW BIAS X -1.8 Y -1.5;
CUT CHECK TO $L3;CCLW BIAS X -.6 Y -.3;
CUT CHECK TO $A8;BIAS X -.6 Y -.3;
RETRACT
SELECT LAY 135
MOVE XYLOC X 0 Y -1.5;
PLANE ZWORK -.375 ZREF
PROFILE TO $A3 $L6 $A2 $L5 $A1 $L4;LEFT CLOSED
PLANE ZWORK -.505 ZREF
PROFILE TO $A3 $L6 $A2 $L5 $A1 $L4;LEFT CLOSED
RETRACT
CLEAR HOME
COOLANT OFF
[DONE]
```

Generating Output

CVNC generates an output file in one of the following formats:

- CLFile
- APT
- COMPACT II

Format type depends on the output processor to be used. For example, the CVGP11 postprocessor requires CLFile format.

In addition to choosing output file format, you can

- Selectively turn output on or off to generate output for certain blocks of the JCF only.
- Turn an auditing facility on that optionally inserts JCF commands into the output file as comment lines. This is useful for solving problems in your JCF.
- Process output in the background, so that output processing will not interfere or slow other processes in CVNC or CADD5.
- Generate output from a previously processed JCF without entering CADD5.

Generating Output for the Mounting Bracket

Because CVGP2 was going to be used to postprocess the output for the mounting bracket part programs, CLFiles were generated. The CVNC OUTPUT command was used to automatically generate a CLFile upon exiting the JCFs. The TEXT modifier created a textual version of the CLFile as well as a binary file. The next three figures show portions of the CLFiles for the bracket's three JCFs. The AUDIT command generates the PPRINT statements in these output files.

CLFile for Opr 1 - Bottom-Mill

```
PPRINT #>*
PPRINT #>* SIDE 1 - OPERATION 1
.
.
.
PPRINT OUTPUT CLFILE TEXT 'BOTTOM-MILL'
PPRINT SELECT LAY 105
PPRINT #>* ZPLANE TOP = 0
PPRINT READ 'TOP' TOPPLANE
PPRINT COMLIB 'SETUP'
PPRINT CALL PLANES
PPRINT CMFIL PLANES
PPRINT PLANE ZREF 0
PPRINT PLANE ZAPPR 1.04 ZREF
PPRINT PLANE ZCLEAR 1 ZREF
PPRINT PLANE ZRETRACT .05 ZREF
PPRINT PLANE ZWORK -.5 ZREF
PPRINT DEFTOOL 'MILL2-1' MILL7 DIA 1 CORNER 0 HEIGHT 3.0
PPRINT HOMEPT FROM X -3 Y 1 Z 5;
FROM -3.0000, 1.0000, 5.0000
PPRINT CLEAR HOME
RAPID
GOTO -3.0000, 1.0000, 5.0000
PPRINT CHGTOOL 3 'MILL2-1'
LOADTL 3 LENGTH 0.0000
PPRINT SPEED RPM 1800 CLW
SPINDL 1800.0000 RPM CLW
PPRINT FEED APPROACH IPM 80PPRINT FEED CLEAR RAPID
PPRINT FEED CONNECT IPM 20
PPRINT FEED CUT IPM 20
FEDRAT 20.0000 IPM
PPRINT FEED PLUNGE IPM 12
PPRINT FEED RETRACT IPM 40
PPRINT COOLANT MIST
COOLNT MIST ON
PPRINT PLANE ZWORK 0 ZREF
PPRINT AREAMILL LACE CUTDIR XAXIS BOUND LOC X -4 Y -4 X 4 Y -4 X 4
Y 2 $
X -4 Y 2;
RAPID
```



```

GOTO -3.0000, 1.0000, 1.0000
GOTO -3.5000, 1.5000, 1.0000
FEDRAT 80.0000 IPM
GOTO -3.5000, 1.5000, 1.0400
FEDRAT 12.0000 IPM
GOTO -3.5000, 1.5000, 0.0000
FEDRAT 20.0000 IPM
GOTO 3.5000, 1.5000, 0.0000
GOTO 3.5000, 0.7500, 0.0000
GOTO -3.5000, 0.7500, 0.0000
.
.
.
GOTO 3.5000, -3.0000, 0.0000
PPRINT RETRACT
FEDRAT 40.0000 IPM
GOTO 3.5000, -3.0000, 0.0500
PPRINT COOLANT OFF
COOLNT OFF
PPRINT CLEAR HOME
RAPID
GOTO -3.0000, 1.0000, 5.0000
    
```

CLFile for Opr 2 - Top-Drill

```

PPRINT #>*
PPRINT #>* SIDE 2 - OPERATION 2
.
.
.
PPRINT OUTPUT CLFILE TEXT 'TOP-DRILL'
PPRINT SELECT LAY 110
PPRINT #>* ZPLANE TOP = 0
PPRINT READ 'TOP' TOPPLANE
PPRINT COMLIB 'SETUP'
PPRINT CALL PLANES
PPRINT CMFIL PLANES
PPRINT PLANE ZREF 0
PPRINT PLANE ZAPPR 1.04 ZREF
PPRINT PLANE ZCLEAR 1 ZREF
PPRINT PLANE ZRETRACT .05 ZREF
PPRINT PLANE ZWORK -.5 ZREF
PPRINT DEFTOOL 'SPOT-2' DRILL DIA .1
PPRINT DEFTOOL 'DR-280' DRILL DIA .280
PPRINT HOMEPT FROM X -3 Y 1 Z 5;
FROM -3.0000, 1.0000, 5.0000
PPRINT CLEAR HOME

RAPID
GOTO -3.0000, 1.0000, 5.0000
PPRINT CHGTOOL 1 'SPOT-2'
LOADTL 1 LENGTH 0.0000
PPRINT SPEED RPM 3600 CLW
    
```

```
SPINDL 3600.0000 RPM    CLW
PPRINT FEED APPROACH IPM 80
PPRINT FEED CLEAR RAPID
PPRINT FEED PLUNGE    IPM 12
PPRINT FEED RETRACT  IPM 40
PPRINT COOLANT MIST
COOLNT MIST    ON
PPRINT NCGROUP HOLESSET ENT $P1 $P2 $P3;
PPRINT DRILL DEPTH .05 ENT HOLESSET;
CYCLE DRILL  RAPTO 0.1000 FEDTO -0.0800 RTRCTO 0.1000 IPM
0.0000
GOTO 0.0000, 0.0000, 0.0000
GOTO -2.0000, -2.0000, 0.0000
GOTO 2.0000, -2.0000, 0.0000
CYCLE OFF
PPRINT RETRACT
FEDRAT 40.0000 IPM
GOTO 2.0000, -2.0000, 0.0500
PPRINT COOLANT OFF
COOLNT OFF
PPRINT CLEAR HOME
RAPID
GOTO -3.0000, 1.0000, 5.0000
PPRINT CHGTOOL 2 'DR-280'
LOADTL 2 LENGTH 0.0000
PPRINT SPEED RPM 3600 CLW
SPINDL 3600.0000 RPM    CLW
PPRINT FEED APPROACH IPM 80
PPRINT FEED CLEAR RAPID
PPRINT FEED PLUNGE    IPM 6
PPRINT FEED RETRACT  IPM 40
```

CLFile for Opr 2 - Top-Drill

```
PPRINT COOLANT MIST
PPRINT COOLANT MIST
COOLNT MIST    ON
PPRINT DRILL DEPTH .6 PECK MAXDEP .2 ENT HOLESSET;
CYCLE DEEP  RAPTO 0.1000 FEDTO -0.6841 INCR 0.1710
RTRCTO 0.1000 IPM    0.0000
GOTO 0.0000, 0.0000, 0.0000
GOTO -2.0000, -2.0000, 0.0000
GOTO 2.0000, -2.0000, 0.0000
CYCLE OFF
PPRINT RETRACT
FEDRAT 40.0000 IPM
GOTO 2.0000, -2.0000, 0.0500
PPRINT COOLANT OFF
COOLNT OFF
PPRINT CLEAR HOME
RAPID
GOTO -3.0000, 1.0000, 5.0000
```

CLFile for Opr 3 - Top-Mill

```
PPRINT OUTPUT CLFILE TEXT 'TOP-MILL'
PPRINT SELECT LAY 120
PPRINT #>* ZPLANE TOP = 0
PPRINT READ 'TOP' TOPPLANE
PPRINT COMLIB 'SETUP'
PPRINT CALL PLANES
PPRINT CMFIL PLANES
PPRINT PLANE ZREF 0
PPRINT PLANE ZAPPR 1.04 ZREF
PPRINT PLANE ZCLEAR 1 ZREF
PPRINT PLANE ZRETRACT .05 ZREF
PPRINT PLANE ZWORK -.5 ZREF
PPRINT DELTOOL 'MILL2-1'
PPRINT DEFTOOL 'MILL2-1' MILL7 DIA 1 CORNER 0 HEIGHT 3.0
PPRINT DELTOOL 'MILL2-2'
PPRINT DEFTOOL 'MILL2-2' MILL7 DIA .375 CORNER 0 HEIGHT 2.0
PPRINT HOMEPT FROM X -3 Y 1 Z 5;
FROM -3.0000, 1.0000, 5.0000
PPRINT CLEAR HOME
RAPID
GOTO -3.0000, 1.0000, 5.0000
PPRINT CHGTOOL 3 'MILL2-1'
LOADTL 3 LENGTH 0.0000
PPRINT SPEED RPM 1800 CLW
SPINDL 1800.0000 RPM CLW
PPRINT FEED APPROACH IPM 80
PPRINT FEED CLEAR RAPID
PPRINT FEED CONNECT IPM 6
PPRINT FEED CUT IPM 8
FEDRAT 8.0000 IPM
PPRINT FEED PLUNGE IPM 12
PPRINT FEED RETRACT IPM 40
PPRINT COOLANT MIST
COOLNT MIST ON
PPRINT MOVE XYLOC X 0 Y -1.5;
RAPID
GOTO 0.0000, -1.5000, 5.0000
PPRINT PLANE ZWORK 0 ZREF
PPRINT AREAMILL LACE CUTDIR XAXIS BOUND LOC X -7 Y -4.5 X 7 Y -4.5
X 0 $
Y 3.5;
RAPID
GOTO 0.0000, -1.5000, 1.0000
GOTO 0.0000, 2.7407, 1.0000
FEDRAT 80.0000 IPM
GOTO 0.0000, 2.7407, 1.0400
FEDRAT 12.0000 IPM
GOTO 0.0000, 2.7407, 0.0000
FEDRAT 6.0000 IPM
```

```
GOTO -0.6562, 1.9907, 0.0000
GOTO 0.6563, 1.9907, 0.0000
GOTO 1.3125, 1.2407, 0.0000

.
.
.
PPRINT RETRACT
FEDRAT 40.0000 IPM
GOTO -5.2500, -3.2593, 0.0500
PPRINT SELECT LAY 125
PPRINT PLANE ZWORK -.2 ZREF
PPRINT AREAMILL LACE CUTDIR XAXIS BOUND LOC X -7 Y -4.5 X 7 Y -4.5
X 0 $
Y 3.5; ISLAND $C
RAPID
GOTO -5.2500, -3.2593, 1.0000
GOTO -5.8981, -4.0000, 1.0000
FEDRAT 80.0000 IPM
GOTO -5.8981, -4.0000, 1.0400
FEDRAT 12.0000 IPM
GOTO -5.8981, -4.0000, -0.2000
FEDRAT 6.0000 IPM
GOTO 5.8981, -4.0000, -0.2000
GOTO 5.2419, -3.2500, -0.2000
FEDRAT 8.0000 IPM
GOTO -5.2419, -3.2500, -0.2000
FEDRAT 6.0000 IPM
GOTO -4.5856, -2.5000, -0.2000
FEDRAT 8.0000 IPM
GOTO -2.8660, -2.5000, -0.2000
FEDRAT 6.0000 IPM
CIRCLE -2.0000, -2.0000, -0.2000
0.0000, 0.0000, -1.0000
1.0000
GOTO -2.8660, -2.5000, -0.2000
-2.9063, -2.4226, -0.2000
-2.9397, -2.3420, -0.2000
.
.
.

CIRCLE 0.0000, 0.0000, -0.2000
0.0000, 0.0000, -1.0000
1.0000
GOTO -1.0000, 0.0000, -0.2000
-0.9962, 0.0872, -0.2000
-0.9848, 0.1736, -0.2000
-0.9659, 0.2588, -0.2000
-0.9397, 0.3420, -0.2000
-0.9063, 0.4226, -0.2000
-0.8660, 0.5000, -0.2000
FEDRAT 8.0000 IPM
GOTO -1.9606, 0.5000, -0.2000
```

```

RAPID
GOTO  -1.9606,  0.5000,  1.0000
GOTO  0.9682,  -1.0000,  1.0000
FEDRAT 80.0000 IPM
.
.
.
2.0173,  -1.0001,  -0.2000
1.9202,  -1.0032,  -0.2000
1.8240,  -1.0156,  -0.2000
1.7294,  -1.0373,  -0.2000
1.6373,  -1.0681,  -0.2000
1.5487,  -1.1076,  -0.2000
1.4643,  -1.1556,  -0.2000
1.3850,  -1.2115,  -0.2000
1.3114,  -1.2748,  -0.2000
1.2444,  -1.3450,  -0.2000
1.1844,  -1.4213,  -0.2000
1.1322,  -1.5031,  -0.2000
1.0881,  -1.5896,  -0.2000
1.0526,  -1.6799,  -0.2000
1.0260,  -1.7733,  -0.2000
PPRINT RETRACT
FEDRAT 40.0000 IPM
GOTO  1.0260,  -1.7733,  0.0500
PPRINT CLEAR HOME

RAPID
GOTO  -3.0000,  1.0000,  5.0000
PPRINT CHGTOOL 4 'MILL2-2'
LOADTL 4 LENGTH 0.0000
PPRINT SPEED RPM 3600 CLW
SPINDL 3600.0000 RPM    CLW
PPRINT SELECT LAY 13
PPRINT MOVE TO $L3;INTOF TO  $A8;BIAS X -.6 Y -.3;
RAPID
GOTO  -0.6803,  -0.2383,  5.0000
PPRINT PLANE ZWORK -.375 ZREF
PPRINT PLUNGE
FEDRAT 12.0000 IPM
GOTO  -0.6803,  -0.2383,  -0.3750
PPRINT CUT TO $L3;INTOF TO  $A8;BIAS X -.6 Y -.3;
FEDRAT 8.0000 IPM
PPRINT CUT CHECK TO $A8;BIAS X -.6 Y -.3;TO $C4;CCLW BIAS X -.55 Y
-.2;$
TO $A5 ;CLW BIAS
CIRCLE -0.7244,  -0.1941,  -0.3750
0.0000,  0.0000,  1.0000
0.0625
GOTO  -0.6802,  -0.2383,  -0.3750
-0.6720,  -0.2282,  -0.3750
-0.6661,  -0.2165,  -0.3750
-0.6627,  -0.2039,  -0.3750
-0.6620,  -0.1909,  -0.3750
    
```

```
-0.6641,   -0.1780,   -0.3750
CIRCLE 0.0000,   0.0000,   -0.3750
.
.
.
0.6641,   -0.1780,   -0.3750
PPRINT CUT CHECK TO $L1;CCLW BIAS X .6 Y -.3;
CIRCLE 0.7244,   -0.1941,   -0.3750
0.0000,   0.0000,   1.0000
0.0625
GOTO 0.6641,   -0.1780,   -0.3750

0.6620,   -0.1909,   -0.3750
0.6627,   -0.2039,   -0.3750
0.6661,   -0.2165,   -0.3750
0.6720,   -0.2282,   -0.3750
0.6803,   -0.2383,   -0.3750
PPRINT CUT CHECK TO $A4;BIAS X 1.65 Y -1.45;TO $C6;CCLW BIAS X 1.8
Y -1$ .5;TO $A7;CLW B
GOTO 1.7617,   -1.3197,   -0.3750
CIRCLE 1.8059,   -1.2756,   -0.3750
0.0000,   0.0000,   1.0000
0.0625
GOTO 1.7617,   -1.3198,   -0.3750
.
.
.
PPRINT PROFILE TO $A3 $L6 $A2 $L5 $A1 $L4;LEFT CLOSED
FEDRAT 12.0000 IPM
GOTO 0.5745,   -1.0165,   -0.5050
FEDRAT 8.0000 IPM
CIRCLE 0.0000,   -1.5910,   -0.5050
0.0000,   0.0000,   1.0000
0.8125
GOTO 0.5745,   -1.0165,   -0.5050
0.4869,   -0.9406,   -0.5050
0.3894,   -0.8779,   -0.5050
0.2839,   -0.8297,   -0.5050
0.1727,   -0.7971,   -0.5050
0.0580,   -0.7806,   -0.5050
.
.
.
GOTO 1.1572,   -1.7500,   -0.5050
CIRCLE 1.1572,   -1.6875,   -0.5050
0.0000,   0.0000,   1.0000
0.0625
GOTO 1.1572,   -1.7500,   -0.5050
1.1855,   -1.7432,   -0.5050
.
.
.
```

```
GOTO 0.5745, -1.0165, -0.5050
PPRINT RETRACT
FEDRAT 40.0000 IPM
GOTO 0.5745, -1.0165, 0.0500
PPRINT CLEAR HOME
RAPID
GOTO -3.0000, 1.0000, 5.0000
PPRINT COOLANT OFF
COOLNT OFF
```

Postprocessing

After a part program has been written and output into CLFile, APT, or COMPACT II format, it must be postprocessed.

Postprocessor software translates an output file into language understood by the machine tool/control unit combination (see figure on the next page). The language that the machine control unit (MCU) understands is called Machine Control Data (MCD), and may be specific for each MCU being used.

Every machine tool has unique characteristics. For example,

- A machine may or may not have an automatic tool changer.
- A machine may have a 60-position rotary index table or a 360-position rotary index table.

Machine control units also have unique characteristics. For example, an MCU may be capable of circular interpolation. If this is the case, the postprocessor will use this capability, overriding the CLFile point data that approximates the arc.

A postprocessor also processes any unprocessed data in the JCF; for example, pass-through statements used for controlling the post or machine tool functions. As defined before, pass-through statements are passed directly to output and do not affect JCF processing.

CVGP II

The CVGP II family of products provides a complete solution for the generation of Machine Control Data, from CVNC CLFiles, for most NC machine tools. CVGP II includes a configuration tool and a postprocessor.

Configuration Tool

The configuration tool allows you to enter machine and controller characteristics specific to a machine. The tool allows you to create a unique parameter file and macros for each machine in your shop. It also supplies a set of default conditions that can be used where appropriate. The configuration tool only needs to be used once for each machine; the postprocessor will use these established parameters whenever executed for that machine.

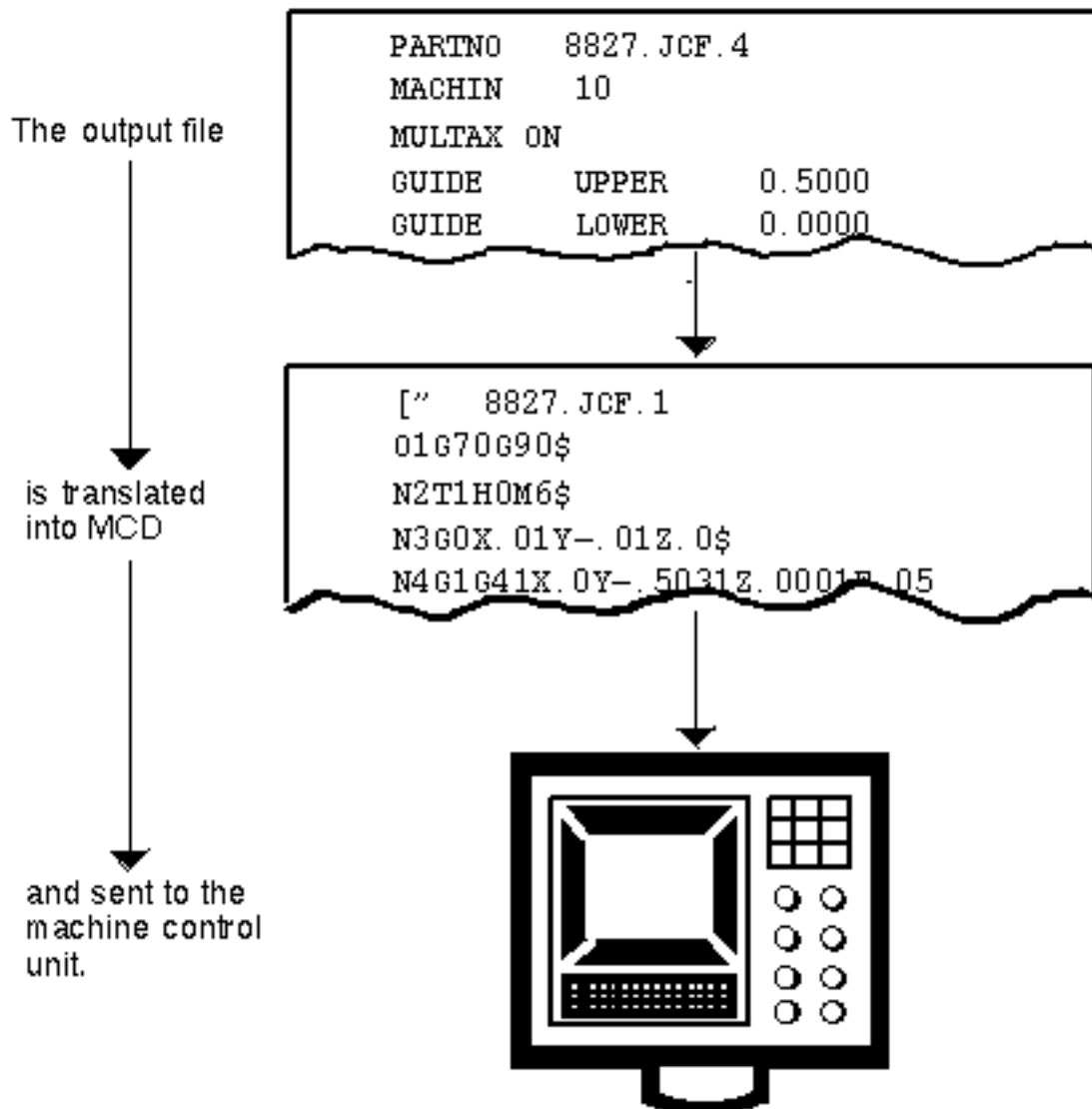
Postprocessor

The postprocessor reads the CLFile and generates two files: machine control data file and an optional operator listing. These files are based upon the contents of the parameter file generated with the configuration tool.

Custom Postprocessors

Custom postprocessors can also be developed to create a postprocessor specifically for your NC machine.

Figure 3-11 Postprocessed Data and the MCU



Postprocessing Output for the Mounting Bracket

The three CLFiles for the mounting bracket JCF were postprocessed using CVGPPII. There are several models of CVGPPII available; use the one that corresponds to the machine tool type you are using. For example, the mounting bracket was machined on a mill machining center without multi-axis capability. It required CVGPPII Model 3580.

Each CVGPPII system consists of a postprocessor and corresponding configuration tool. CVGPPII-M2 includes the configuration tool command `mpg2` and the postprocessor command `mill2p`. `mpg2` and `mill2p` are operating system-level commands.

To postprocess the mounting bracket CLFiles, the following steps were followed. Step 1 only needs to be performed once for a machine (unless machine parameters need to be changed). With subsequent CLFiles, only step 2 must be performed.

1. Execute the configuration tool.

The configuration tool asks for the name and number of the machine tool. It creates a file of data for that machine that is interpreted by the generalized postprocessor at execution time. The configuration tool allows you to change any one of more than 500 specifications used to define such things as

- Letter Address Codes
- Register Formats
- Machine and Controller Names
- Preparatory G-codes
- Auxiliary M-codes

The specifications are broken into 18 categories that are identified by a number and can be selected in any order (see “CVGPPII Configuration Tool Main Menu” on page 3-44). If you do not select and alter a category, standard defaults are used.

2. Execute the postprocessor.

If a `MACHIN/name,number` is found in the CLFile that corresponds with a configuration file that you created with that same name and number, the postprocessor will process the CLFile using the characteristics as defined in that file. The figure on the next page, *Printing the Processed CLFile to the Terminal with mill2p*, shows the processed CLFile printed to the screen.

CVGPPII Configuration Tool Main Menu

Automation Intelligence, Inc. Postprocessor Configuration Tool

Machine No. /mill/1

POSTPROCESSOR CATEGORIES :

| | |
|----------------------------------|--------------------------------|
| 1 = POSTPROCESSOR NAME | 10 = DWELL SPECIFICATIONS |
| 2 = COORDINATE SYSTEM DEFINITION | 11 = SPINDLE INFORMATION |
| 3 = MACHINE TYPE SELECTION | 12 = FEEDRATE INFORMATION |
| 4 = HOME ORIGIN & TRAVEL LIMITS | 13 = TOOL DEFINITION AND USAGE |
| 5 = LETTER ADDRESS DEFINITION | 14 = CIRCULAR INTERPOLATION |
| 6 = REGISTER FORMATS | 15 = TAPE CONTROL FEATURES |
| 7 = OUTPUT ORDER SELECTION | 16 = MACRO DEFINITION |
| 8 = G CODE SELECTION | 17 = PP VOCABULARY WORDS |
| 9 = M CODE SELECTION | 18 = ALL OTHER SPECIFICATIONS |

Enter X to exit

PGP > ENTER CATEGORY :

Printing the Processed CLFile to the Terminal with mill2p

***** AUTOMATION INTELLIGENCE, INC. *****
GENERALIZED POSTPROCESSOR (VERSION 4 MOD 03)
MACHINE NUMBER: 1.0 DATE: 30 Jul 89 PAGE: 1

```
#>*
#>* SIDE 2 - OPERATION
#>
OUTPUT CLFILE TOP-MILL
DEFTOOL MILL2-1 MILL7 DIA 1 CORNER 0 HEIGHT 3.0
DEFTOOL MILL2-2 MILL DIA .375 CORNER 0 HEIGHT 2.0
HOMEPT FROM X -3 Y 1 Z 5;
  N1 G70 G90
    FROM -3.00000 1.00000 5.00000
CLEAR HOME
CHGTOOL 3 MILL2-1
  O2 T3 H0 M6
.
.
.
N999 MO9
```

Distributing Data

The next step in the NC solution is the distribution of data to the machine control unit. With CVNC you can use two distribution mediums: tape and direct numerical control hardware.

Paper Tape

Postprocessed data can be distributed to a Machine Control Unit via paper tape. Data can also be retrieved from tape and stored in a text file for comparison with original data and editing.

Direct Numerical Control

Direct numerical control hardware and software connects a group of numerically controlled machines to a common computer system. The direct numerical control host computer

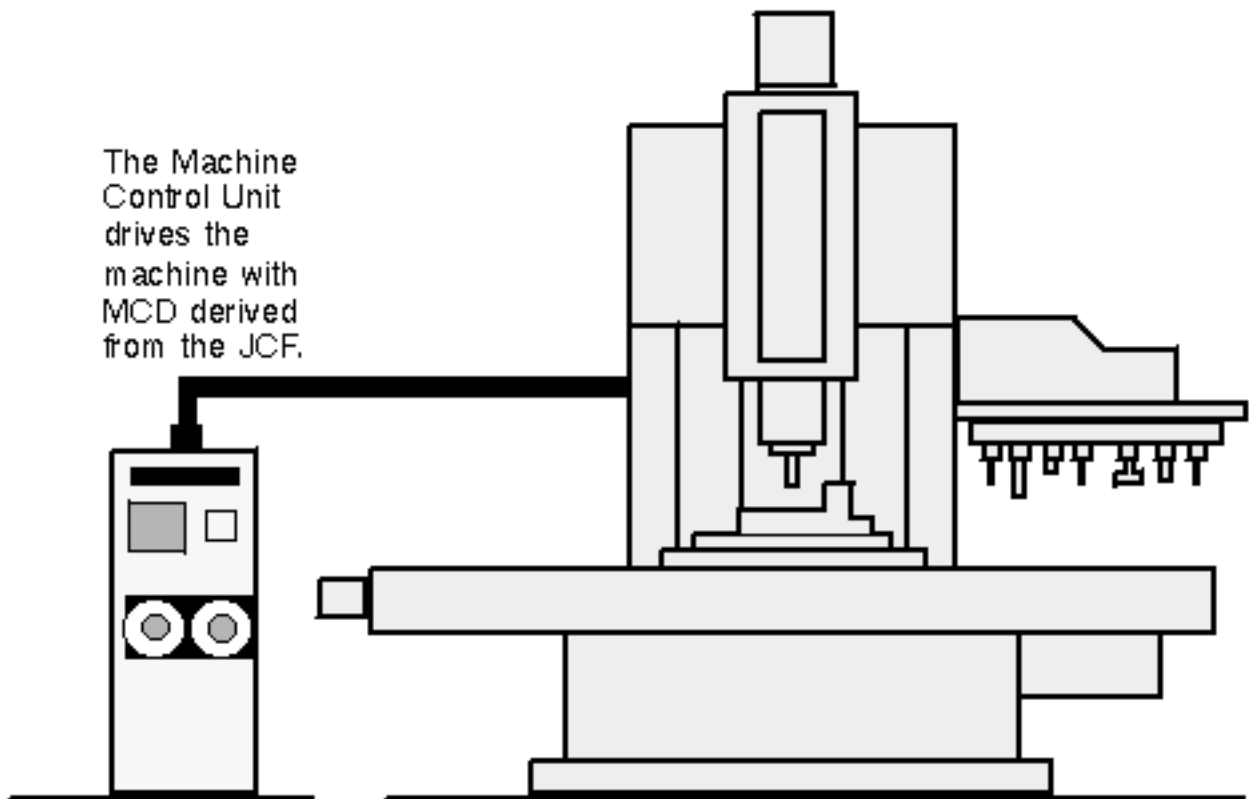
- Displays and edits part programs
- Transmits operator instructions
- Collects and processes machine status and other information needed throughout the factoryMachining

The Machine Control Data drives the machine tool to machine the part. The Machine Control Unit (MCU) is the link between the part programmer and the machine tool. After the programmer writes a JCF with CVNC, generates output, postprocesses the output to create MCD, and distributes the MCD to the MCU, machining can begin.

The MCU converts the MCD into machine motions through electrical or hydraulic servomechanisms. MCUs are programmed to control the type of machine to which they are attached. The physical components on a lathe's MCU are similar to the components on a milling machine's MCU; it is the way the MCU is programmed that makes MCUs different on various machines.

Generally, the MCU is programmed by the MCU manufacturer and this program is not changed by the machine operator. The MCU does have memory allocated to additional programming by the machine builder to adapt the MCU to a particular machine. Almost all MCUs allow instructions to be manually entered at the control.

Please note: MCD can also be read into NC machines without programmable MCUs; CVNC supports these machines.



Index

A

- Application commands 3-9
 - job setup
 - machine configuration 3-10
 - output generation 3-13
 - tool definition 3-12
 - operation setup 3-15
 - tool motion 3-19
- APT
 - output
 - sample 2-29

C

- CADDS
 - accessing from CVNC 2-12
 - geometry referencing 1-15
 - integration with CVNC 1-15, 1-18
 - part design considerations 3-2
- Check macros 2-26
- CLFile
 - output
 - sample 2-28
- Command files
 - decomposing 2-24
 - sample 2-25
 - uses for 1-22
- Command macros 2-26
- COMPACT II
 - output
 - sample 2-30
- Construction planes 3-6, 3-15

- Customizing
 - CVNC 1-22
- Cutting motion of tools 3-19
- CVGPll (Postprocessor) 1-18, 3-42
- CVMAC
 - integration with CVNC 1-18
 - macros 1-22
 - decomposing 2-24
 - description 2-26
 - JCF function of 3-8
 - types of 2-26
- CVNC overview
 - base system 1-2
 - command files 1-22
 - command structure 1-12
 - concepts 1-6
 - customizing 1-22
 - desktop 1-19
 - documents 1-24
 - editor 2-17
 - sample of commands 2-17
 - environment management 1-19
 - execute files 1-22
 - family of products 1-2
 - graphic display 1-12
 - information levels 1-13
 - language 1-11
 - macros 1-22
 - menus and icons 2-5
 - property sheets 2-8
 - system variables 1-22
 - user interface 1-19
- CVNC-M2
 - cutting motion 3-19
 - features 2-31

- descriptions 2-34
- examples (illustrated) 1-7
- integration with CVNC-M3 2-39
- CVNC-M3
 - cutting motion 3-19
 - features 2-31
 - descriptions 2-39
 - examples (illustrated) 1-8
 - integration with CVNC-M2 2-39
- CVNC-M5
 - 4-axis machining 2-54
 - cutting motion 3-20
 - features 2-32
 - descriptions 2-51
 - examples (illustrated) 1-9
- CVNC-T2
 - cutting motion 3-20
 - features 2-32
 - descriptions 2-55
 - examples (illustrated) 1-10

D

- Data distribution to MCU 3-46
- Decomposing
 - commands
 - example 1-13, 2-24
- Direct Numerical Control (DNC) 3-46
- Documentation, printing from Portable Document Format (PDF) file xvi
- Drawing views 3-5

E

- Entities
 - grouping
 - in operation setup 3-16
 - uses for 2-16
 - selecting
 - by digitizing 1-15
 - by name (tag) 1-16
 - tagging (naming) 1-16
- Environments
 - desktop access to 1-19
 - functions 1-20
- Execute files 2-25

- uses for 1-22

H

- Help
 - accessing online 2-9
- Hole processing 2-38

I

- Isoparameters 2-40

J

- JCFs 2-13
 - characteristics 1-12
 - contents 3-8
 - application commands 3-9
 - system commands 3-9
 - creating 3-8
 - example (sample part) 3-21
 - methods of 2-13
 - editing 2-17
 - sample of commands for 2-17
 - editing commands for 2-13
 - output files from 2-27
 - commands for 2-28
 - samples
 - APT format 2-29
 - CLFile format 2-28
 - COMPACT II format 2-30
 - sample 2-14
 - tags and 1-16
 - geometry changes and 1-16
- Job setup commands
 - machine configuration 3-10
 - output generation 3-13
 - tool definition 3-12

M

- Machine configuration 3-10
- Machine Control Data (MCD) 3-46
 - postprocessing and 3-42

Machine Control Unit (MCU) 3-46
 postprocessing and 3-42
 example (illustrated) 3-43
 Macros 1-22
 MCD 3-46
 MCU 3-46
 Menus and icons 2-5

N

Naming
 entities 1-16
 NC process
 functions 1-7
 steps 1-4
 NC Tool Path Verifier 2-22
 NCGROUPS
 creating
 example 2-16
 in operation setup 3-16
 uses for 2-16

O

Online documentation 2-9
 Operation setup commands 3-15
 Output 2-27
 APT 3-33
 CLfile 3-33
 COMPACT II 3-33
 formats 3-13
 pass-through statements 1-23
 processing
 generating output 3-33
 postprocessing 3-42
 specifying output format 3-13
 Output macros 2-26

P

Pass-through statements 1-23
 JCF function of 3-8
 Point macros 2-26
 Postprocessors

custom 3-43
 CVGPIL 3-42
 data distribution to MCU 3-46
 MCU, MCD and 3-42
 requirements 3-33
 using 3-42
 PPRINT statements 3-34
 Printing documentation from Portable
 Document Format (PDF) file xvi
 Property sheets 2-8
 Punch tools 2-59
 shapes 2-59

S

Sample part
 description 3-3
 introduction 1-26
 JCF creation 3-21
 output generation 3-34
 postprocessing 3-44
 Setup commands
 job setup
 for machine configuration 3-10
 for output generation 3-13
 for tool definition 3-12
 operation setup 3-15
 Star buffers
 editor commands stored in 2-17
 Status
 displays
 customizing 1-22
 system variables and 1-22, 2-15
 Surface lists
 entity grouping with 2-16
 using 2-46
 Swarfcutting 2-52
 System variables 2-15
 displaying 1-22
 displaying status of 2-15
 types of 2-15

T

Tags
 entity groups and 2-16

- geometry changes and 1-16
- part program (JCF) function of 1-16
- specifying entities with 1-16
- Tool paths
 - display 2-20
 - JCFs and 3-8
 - sample 2-21
 - updating graphics of 2-17
 - verifying 2-22
 - sample 2-23
- Tools
 - 7-parameter 2-35, 2-49
 - cutting and noncutting movement 3-19
 - defining 3-12
 - displaying 2-19
 - milling types 2-34
 - movement of 3-19
 - path verification 2-22
 - property sheets for 2-9
 - selection 3-15
 - turning types 2-57

U

- User interface 2-3
 - desktop layout (diagram) 2-4
 - editing facilities from 2-5
 - left panel functions of 1-20
 - menus and icons 2-5
 - property sheets 2-8