

CVNC™ Editor Guide

CADD5®5 Revision 8.1.2

DOC38527-005

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	xi
Book Conventions	xii
Window Managers and the User Interface	xiii
Online User Documentation	xiii
Online Command Help	xiv
Printing Documentation	xiv
Resources and Services	xv
Documentation Comments	xv

Overview and Quick Reference

Overview	1-2
Graphics Updates	1-2
CVNC Editor Facilities	1-2
CVNC Editor Versus Text Editors	1-2
Advanced CVNC Editor Facilities	1-2
CVNC Editor Terms	1-3
Getting Started	1-4
Using the CVNC Editor Icons	1-4
Using the Keyboard	1-4
Using the Editor Popup Menu	1-6
Quick Reference	1-7

Getting CVNC Editor Online Help

Using the Editor Assist Lines _____	2-2
Using the ! _____	2-3

Working with the JCF

Displaying JCF Lines — P and T _____	3-2
At the Top of the JCF _____	3-2
Within the JCF _____	3-2
Stepping Forward and Backward — F, B, and RESTART _____	3-4
Moving through the JCF _____	3-4
Inserting New JCF Lines _____	3-4
Stepping Forward — F _____	3-5
F Options _____	3-5
Stepping Backward — B _____	3-5
B Options _____	3-6
Returning to the JCF Top — RESTART _____	3-6
Ending Your JCF Session — EXIT and QUIT _____	3-8
Controlling the Display — SEE, BR, FU, and GR _____	3-9
Displaying Tags and Location Data — SEE _____	3-9
Long Mode _____	3-9
Short Mode _____	3-9
Editing with ED or E _____	3-10
Searching for Strings _____	3-10
SEE Examples _____	3-10
Suppressing JCF Line Printing — BR and FU _____	3-10
Viewing Updated Graphics Displays — GR _____	3-11
Managing Changes to Existing JCFs - LOCK, UNLOCK, FC, and RECOVER _____	3-13
Locking a Command - LOCK _____	3-13
LOCK Options _____	3-14
Unlocking a Command - UNLOCK _____	3-14
UNLOCK Options _____	3-14

Comparing Evaluations - FC_____	3-14
FC Options_____	3-15
The COMPARE Command_____	3-15
Recovering the Original Evaluation - RECOVER_____	3-16
Saving JCF Contents — SECURE, SAVE, and RESTORE _____	3-17
SECURE Procedure_____	3-17
SAVE Procedure_____	3-19
RESTORE Procedure_____	3-19
Controlling Processing — SYN, ON, OFF, and EX _____	3-21
CVNC Command Processing and Syntax Checking _____	3-21
Turning Command Processing/Syntax Checking Off — OFF _	3-21
Turning Syntax Checking On/Processing Off — SYN _____	3-21
Turning Command Processing/Syntax Checking On — ON __	3-21
Turning Command Execution On and OFF — EX _____	3-22
Packing the JCF — PACK and PACK JCF_____	3-24
CVNC PACK Procedure_____	3-24
CADD5 PACK Procedure_____	3-24

Editing JCF Lines

Editing Commands Overview _____	4-2
Editing JCF Lines_____	4-2
Deleting Line Information Before it is Processed_____	4-2
Copying and Moving JCF Lines_____	4-2
Editing a Line — E _____	4-4
Editing Example _____	4-4
Editing Graphic Entities — ED _____	4-6
Substituting Text — C _____	4-7
Replacing Text _____	4-7
Prefixing Text _____	4-7
Deleting Text _____	4-7
C (Change) Procedures _____	4-7
Creating Comment Lines — #>_____	4-9
Appending Text — +_____	4-10

Deleting Lines — D _____	4-11
D Options _____	4-11
Canceling a Deletion — UNDEL _____	4-12
Repeating Command Lines — A and !! _____	4-13
Repeating Editor Commands — A _____	4-13
Marking and Unmarking Lines — . and - _____	4-15
Marking a Line _____	4-15
Removing a Mark _____	4-15
Selecting and Copying Lines — # and = _____	4-16
Cutting and Pasting Lines — < and > _____	4-19

Accessing CADD5 from CVNC

Entering Single CADD5 Commands — ~ and IC _____	5-2
Accessing CADD5 — ~ and IC _____	5-2
Moving to CADD5 with the ~ _____	5-2
Moving to CADD5 with IC _____	5-3
Entering Multiple CADD5 Commands — RESU _____	5-4
Moving to CADD5 with RESU _____	5-4
Returning to CVNC with RESUME NCPROC _____	5-4

Automating JCF Input

CVNC Command and Execute Files _____	6-2
Performing Repetitive Functions _____	6-2
Choosing Which Facility to Use _____	6-3
Command and Execute File Commands _____	6-3
Using Command Files _____	6-4
Storing Command Files _____	6-4
Command File Contents _____	6-4
Command File Creation Methods _____	6-5
Creating Online Command Files _____	6-5
Creating Offline Command Files _____	6-7
Accessing Command Files _____	6-7

Modifying Command Files _____	6-8
Locating Lines within a Command File _____	6-8
Using Execute Files _____	6-9
Execute File Storage _____	6-9
Execute File Contents _____	6-9
Creating an Execute File _____	6-9
Accessing Execute Files _____	6-10
Modifying an Execute File _____	6-10

Decomposing JCF Lines

Using Command	
Decomposition - IN, INX,), and OUT _____	7-2
Decomposition — IN _____	7-2
Recomposition — OUT _____	7-2
Decomposing Motion Commands _____	7-3
Using IN _____	7-3
FEED/CUT Format Example _____	7-4
Motion Format Example _____	7-4
Combined FEED/CUT and Motion Format Example _____	7-5
Special Considerations with TRANSLATE, ROTATE, and MIRROR	7-6
Decomposing Command Files _____	7-7
Decomposing Command Files with IN (or INX) _____	7-7
Decomposing CVMAC Macros _____	7-9
Decomposing Macros with INX _____	7-9
Decomposing Macros with IN _____	7-10

Using CVNC Search and Editing Facilities

Search and Editing Facilities _____	8-2
Star Buffers _____	8-2
Metacharacters _____	8-2
Star Buffers and Metacharacters _____	8-3

Using Metacharacters _____	8-4
Metacharacter Symbols _____	8-4
Metacharacter Examples _____	8-4
Using Star Buffers _____	8-7
Star Buffer Procedure _____	8-7
Star Buffer Commands _____	8-7
Creating a Star Buffer — *n _____	8-8
Listing Star Buffers — L* _____	8-8
Displaying Star Buffers — T*n _____	8-9
Executing a Star Buffer — X*n _____	8-9
X*n Options _____	8-10
Editing a Star Buffer — E*n _____	8-10
Inserting Additional Commands — I*n and IL _____	8-11
Saving Star Buffer Edits — FILE* _____	8-12
Discarding Star Buffer Edits — QUIT* _____	8-13
Combining Metacharacters and Star Buffers _____	8-15
Star Buffer and Metacharacter Example _____	8-15
Displaying Search Strings (TW) _____	8-16

Preface

The *CVNC Editor Guide* provides instructions and reference information on all CVNC editor commands. This book is written for NC part programmers, operators, and manufacturing engineers using CVNC for the production of control tapes for numerically controlled machine tools.

Related Documents

The following documents may be helpful as you use the *CVNC Editor Guide*:

- *Understanding CVNC*
- *CVNC System User Guide and Menu Reference*
- *CVNC System Variables Guide*
- *Customizing CVNC*
- *CVNC Work Examples*
- *CVNC Master Index*

Book Conventions

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

Convention	Example	Explanation
Menu selections and options	List Section option, Specify Layer field	Indicates a selection you must make from a menu or property sheet or a text field that you must fill in.
User-selected graphic location	X, d ₁ or P1	Marks a location or entity selection in graphic examples.
User input in CADDs text fields and on any command line	cvaec.hd.data.param tar -xvf /dev/rst0	Enter the text in a CADDs text field or on any command line.
System output	Binary transfer complete.	Indicates system responses in the CADDs text window or on any command line.
Variable in user input	tar -cvf /dev/rst0 filename	Replace the variable with an appropriate substitute; for example, replace filename with an actual file name.
Variable in text	tagname	Indicates a variable that requires an appropriate substitute when used in a real operation; for example, replace tagname with an actual tag name.
CADDs commands and modifiers	INSERT LINE TANTO	Shows CADDs commands and modifiers as they appear in the command line interface.
Text string	"SRFGROUPA" or 'SRFGROUPA'	Shows text strings. You must enclose text string with single or double quotation marks.
Integer	n	Supply an integer for the <i>n</i> .
Real number	x	Supply a real number for the <i>x</i> .
#	# mkdir /cdrom	Indicates the root (superuser) prompt on command lines.
%	% rlogin remote_system_name -l root	Indicates the C shell prompt on command lines.
\$	\$ rlogin remote_system_name -l root	Indicates the Bourne shell prompt on command lines.

Window Managers and the User Interface

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

Look and Feel of User Interface Elements

User Interface Element	Common Desktop Environment (CDE) on Solaris, 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 CADDs desktop or the Local Data Manager (LDM)

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

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

From an HTML Browser:

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

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

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

From the Documentation CD-ROM:

1. Mount the documentation CD-ROM.
2. Point your browser to:
CDROM_mount_point/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 CADDSS desktop in the following ways:

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

From the Information Access Button on the 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 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*.

Overview and Quick Reference

This chapter provides general information on the CVNC editor. The chapter also describes how to use the CVNC interface selections to enter editor commands directly from the keyboard or use the ? switch to access editor commands.

- Overview
- Getting Started
- Quick Reference

Overview

The CVNC editor allows you to modify commands and command sequences in your JCF. For example, you can step backward and forward in the JCF to correct, insert, or delete commands. Command input is the default mode in CVNC.

Please note: CVNC editor commands do not become a part of the JCF. They are only used to change or manipulate it.

Graphics Updates

CVNC handles graphics changes immediately. For example, you can step backward in your JCF; watch the associated tool path and tool graphics disappear; make changes with the editing commands; and step forward over these changed commands. Graphics are updated accordingly.

If you modify motion commands, moving forward over them may result in different tool path graphics. Note that you do not delete the lines over which you back up. Thus, reprocessing does not require you to reenter these lines.

CVNC Editor Facilities

A range of advanced facilities are built on the two basic actions, forward and backward, along with the ability to insert or delete a line in your JCF. These advanced facilities allow you to manipulate individual JCF lines as well as groups of JCF lines. As a result, you can evolve your final machining strategy by examining graphics and modifying commands if necessary.

CVNC Editor Versus Text Editors

CVNC editor functions are similar to those of many text editors and result in edited text comprising JCF commands. However, in CVNC, while editing your JCF, you are simultaneously modifying the NC part program.

Advanced CVNC Editor Facilities

The star buffer in the CVNC editor allows you to store groups of editor commands in sequence and then repeatedly call and execute them in your JCF. See Chapter 8, “Using Star Buffers” for detailed description and examples of star buffer commands.

CVNC Editor Terms

Terms used in conjunction with the CVNC editor include:

Line: Defines a line in a JCF as:

- An application statement (command, macro, or pass-through statement). For example,

```
1      PLANE ZAPPR -3
```

- A comment. For example,

```
1      #> THIS JCF BELONGS TO RICH
```

Current line: Defines the line about to be executed. The current line is surrounded by attention markers, indicated by a right angle bracket (>) and a left angle bracket (<). For example,

```
1      > DATUM 'TOP' <
```

The current line can be edited using CVNC editor commands.

Inactive lines: The left square bracket ([) and right square bracket (]) enclose inactive JCF lines, such as the header line. For example,

```
1      [ EOF ]
1      [ DONE ]
```

Nonexecutable lines: Parentheses (()) enclose lines that have been made nonexecutable. You can create nonexecuting lines with the EX command (see Chapter 3, “Controlling Processing — SYN, ON, OFF, and EX.”).

String: Describes a group of alphanumeric characters that can be specified as part of the following CVNC editor commands: F (Forward), B (Backward), C (Change), and + (Append).

System markers: Identifies information in your JCF, such as blocks of commands. These markers are created and displayed by the system. For example, .Top., .Bottom., and continue.

Text delimiter: Encloses text strings in the following editor commands: F (Forward), B (Backward), C (Change), and + (Append). A text delimiter can be a slash (/), comma (,), or semicolon (;). The particular use of the delimiter is specified in the relevant command description.

Getting Started

Using the CVNC Editor Icons

Use the CADD5 CVNC interface icon selections, enter editor commands directly from the keyboard, use the ? switch to access eight editor commands, or use a combination of these methods to edit your JCF.

Look for CVNC editor icons in a task set to the immediate right of the milling task set icons. For general information on using the CADD5 CVNC interface, see the *CVNC System User Guide and Menu Reference*.

The following page shows the editor icons, along with their corresponding commands.

To use the editor icons, highlight your selection, and, if necessary, add more information from the keyboard.

The editor commands are displayed in the CADD5 text window, but do not become part of the JCF.

















Please note: The editor icons contain the most commonly used editor commands. There are some additional editor commands described in this book that you must enter from the keyboard. Do not include CVNC editor commands within CVMAC macros. CVNC Batch does not support editor commands.

Using the Keyboard

As with all CVNC commands, you may enter editor commands into the CADD5 text window at the NC:> prompt from your keyboard. For example,

```
NC:> F/DONE
```

Figure 1-1 Editor Icons

	Moves the editor strip Contains window options		D - Deletes the current line
	Tilde - Provides CADD5 access		P20 - Displays 20 JCF lines
	SEE - Switches between two display modes for tag and location data		P - Displays five JCF lines
	C/ - Replaces text		B/ - Steps back to the specified string
	ED - Allows a single entity or a group of entities to be redigitized		B - Steps back one line
	E - Enters edit mode		F/ - Steps forward to the specified string
	GR - Switches between two modes for updating frequency of graphics		F - Steps forward one line
	UNDEL - Cancels a previous line deletion		EX - Switches execution of the current line off and on

Using the Editor Popup Menu

By default, when you enter a JCF, the ? switch is turned off. This allows you to select eight editor commands from a popup menu (as shown on the next page) by pressing the right mouse button while the cursor is in the CADDs graphics window.

The bottom selection on the popup menu (?) turns the switch on. When the switch is on, the application command and pass-through statement options available are displayed in alphabetical order.

You can select the application command and pass-through statement options in the same manner as the editor commands. For more information, see the *CVNC System User Guide and Menu Reference*.

Figure 1-2 Editor Popup Menu

editor
P50
P
F
B
D
E
A
?

Quick Reference

Command	Purpose
Symbols	
!	Displays CVNC editor command information in the documentation window.
#!	Displays the system variables listing in the documentation window.
command!	Displays online help in the documentation window for the specified command name.
?	Switches online help in the status window and the popup menus on and off.
~	Moves to CADDs level from within a CVNC session.
!!	Repeats the application command you have just entered in your JCF and makes it the new current line.
#	Selects JCF lines to be copied.
#>	Identifies as a comment any alphanumeric text that follows a hash mark (#) followed by a right angle bracket (>) in columns 1 and 2.
+/string	Adds a specified text string to the end of the current line.
-	Removes a file mark from the current line.
.	Places a file mark on the current line.
/	Locates a specified string in the JCF at any nested level.
<	Selects a line(s) to be cut from the preceding file mark to the one before the current line.
=	Inserts before the current line the line(s) selected with the # (Select) command.
>	Specifies the destination for the line(s) previously cut using the < (Cut) command.
[Prepares a block of JCF lines.
]	Mends a call to a high-level command (such as AREAMILL), a command file, or a macro.

Command	Purpose
Alphabetical	
A	Reexecutes a CVNC editor command.
B	Moves the pointer backward one line.
Bn	Moves the pointer backward the specified number of lines.
B.	Moves the pointer backward to the last file mark.
B/string	Moves the pointer backward to the specified string of characters.
B:	Moves the pointer back to the command that generated the digitized graphic tool path segment.
BR	Suppresses lengthy output to the screen when you move forward or backward in your JCF.
F.	Moves the pointer forward to the next file mark.
F/string	Moves the pointer forward to the specified string of characters.
FC	Compares the old and new tool paths for the current line in the JCF and displays the differences in the form of text or graphics.
FCn	Compares the specified number of lines from the current line.
FC.	Compares from the current line up to the next file mark.
FC/string	Compares from the current line up to the specified string.
FU	Restores full output to the screen when you move forward or backward in your JCF.
GR	Switches between two modes for frequency of graphics updating.
IC	Allows you to enter a single CADDs command within your CVNC session.
IN	Enters the next nested level.
INX	Enters the next nested level. (Unlike IN, INX executes commands that have not been executed before going into them. INX also recognizes the lowest command level and does not perform any other action at this level.)
LOCK	Locks evaluation of the current line in the JCF.
LOCKn	Locks evaluation of the specified number of lines from the current line.

Command	Purpose
LOCK/string	Locks evaluation of the JCF from the current line up to the specified string.
ON	Specifies syntax checking and execution of command lines.
OFF	Turns off syntax checking and execution of command lines.
OUT	Moves forward to the first line of the next level up.
P	Displays five JCF lines.
Pn	Displays lines forward from the specified number.
P:	Displays lines identified by a digitize.
PLn	Displays N active lines.
PACK	Removes wasted space from the binary JCF files (<code>jcfname._jcf</code> and <code>jcfname._cmp</code>). You can also use the CADDs PACK command for this task.
QUIT	Returns you to the CADDs prompt, restores the CADDs database and graphics to the state they were in when you entered CVNC (except for inserted entities and tags), and does not file this version of your JCF.
RECOVER	Recovers the earlier evaluation of commands.
RESTART	Positions the pointer to the first line in the JCF.
RESTORE	Replaces the current JCF with a previous JCF of the same name that was stored in the <code>jcfsave/_bcd</code> directory using SAVE.
RESU	Moves to CADDs level from within a CVNC session. (Enter RESUME NCPROC to return to CVNC.)
SAVE	Saves a text JCF under the <code>jcfsave</code> directory.
SECURE	Saves your JCF contents and CADDs part during a work session.
SEE	Switches between two display modes for tag and location data. Enter SEE to switch from long mode to short mode. Enter SEE again to return to long mode.
SYN	Specifies syntax checking without execution.
T	Displays the current line.

Command	Purpose
Alphabetical - cont	
T:	Performs a type operation based on the line identified by a digitize.
TL	Displays the next active line.
TLn	Displays a specified number of active lines.
Tn	Displays a specified number of lines starting with the current line.
TW	Displays the contents of the last text string located by the editor.
UNDEL	Restores the lines flagged for deletion.
W	Writes a block of commands in a JCF to an external file specified with the COMLIB command.
w,filename	Writes the newly created command file to the specified file name.
UNLOCK	Unlocks the current line in the JCF and enables evaluation.
UNLOCKn	Unlocks the specified number of lines from the current line.
UNLOCK/string	Unlocks the JCF from the current line up to the specified string.
Star Buffer Commands	
*n	Initiates star buffer mode.
E*n	Initiates an editing session of a previously defined star buffer.
FILE*	Returns you to normal mode and saves all changes made during star buffer edit mode.
I*	Initiates star buffer input mode.
IL	Inserts a line of application command where text is the text of the application command you are inserting into the JCF.
L*	Displays a list of star buffers associated with the JCF.
QUIT*	Returns you to normal mode without saving any of the changes made during star buffer edit mode.
T*n	Types out the contents of the star buffer with index number n.
X*n	Executes a star buffer with index number n.
X*n,m	Executes a star buffer with the index number n (in the range 0 to 9) a specified number of times (m).
X*n,\$	Executes a star buffer with the index number n (in the range 0 to 9) repeatedly until reaching a condition that will not allow further execution (for example, [DONE]).

Getting CVNC Editor Online Help

This chapter describes how to obtain editor online help from assist lines or using the ! to access online documentation.

- Using the Editor Assist Lines
- Using the !

Using the Editor Assist Lines

If you are using the CADD5 CVNC interface, you can obtain editor online help from the Information Access icon (i) on the top bar menu in the CADD5 CVNC interface.

To display online CVNC editor assist lines,

1. Choose the Information Access icon (i) from the top bar menu.
2. Choose ASSISTANCE.

The system displays assistance lines for each editor command above the CADD5 text window.

As you move the cursor to each editor icon, the assist line for that command appears. For example,

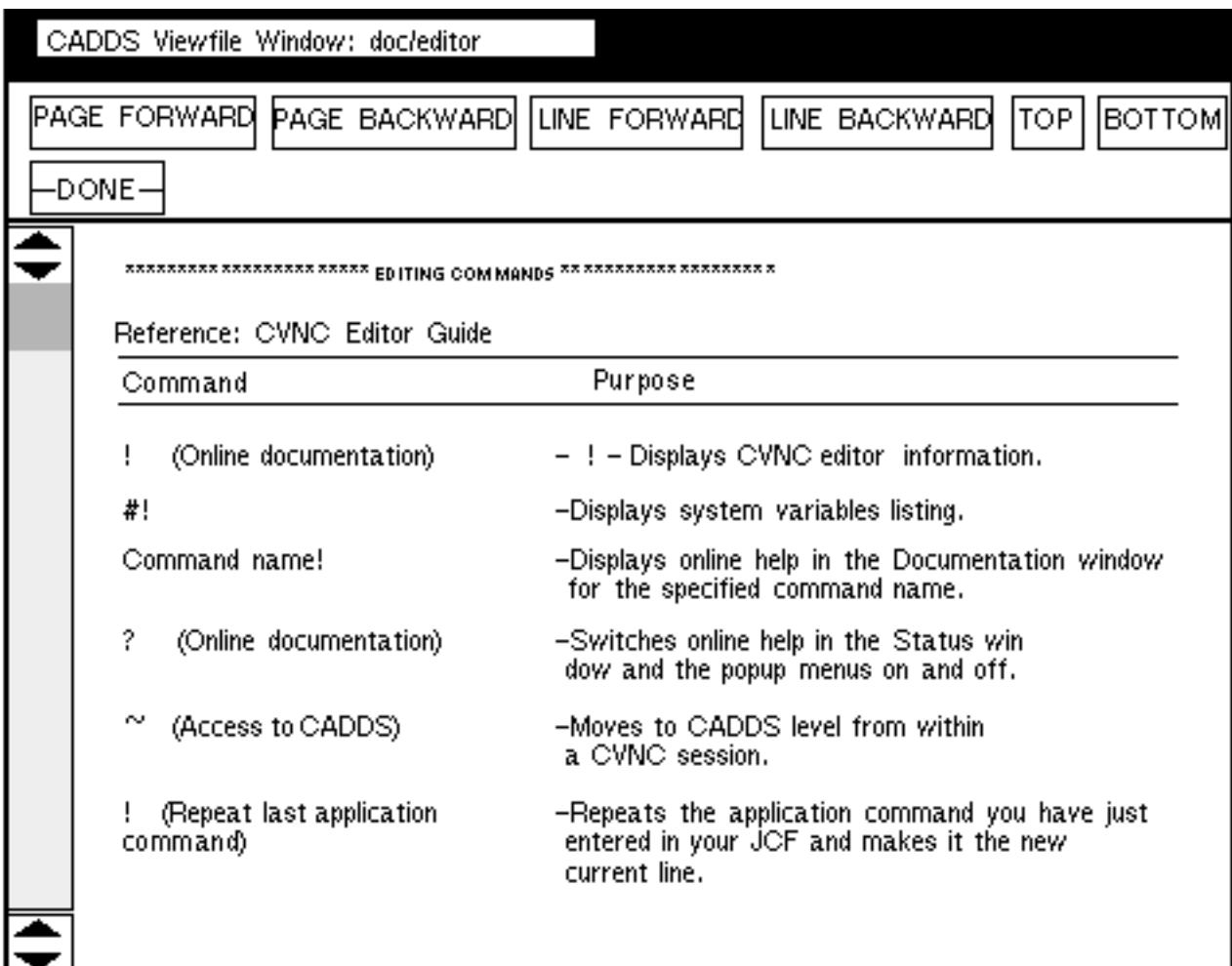
```
Assist: ~ - Moves to CADD5 level from within a  
          CVNC session.
```

Using the !

To see CVNC editor command information, enter ! at the NC :> prompt. CVNC editor command information is displayed in the documentation window. For example,

```
Building documentation window for
=USR.APL.CADDS.DATA.CAM.NCPROC.DOC.&BCD.
EDITOR
```

Figure 2-1 CADDS Viewfile Window



Working with the JCF

This chapter describes how to display JCF lines, move forward and backward in the JCF, manage changes in existing JCF's, exit CVNC and save your JCF changes, quit CVNC and discard any JCF changes made during the session, save the current state of your JCF and part without exiting CVNC, control automatic syntax checking and command processing, and remove wasted space from your JCF using the PACK command from CVNC or the CADDs PACK command.

- Displaying JCF Lines — P and T
- Stepping Forward and Backward — F, B, and RESTART
- Ending Your JCF Session — EXIT and QUIT
- Controlling the Display — SEE, BR, FU, and GR
- Managing Changes to Existing JCFs - LOCK, UNLOCK, FC, and RECOVER
- Saving JCF Contents — SECURE, SAVE, and RESTORE
- Controlling Processing — SYN, ON, OFF, and EX
- Packing the JCF — PACK and PACK JCF

Displaying JCF Lines — P and T

To display JCF lines, use the P or T command. P displays lines above and below the current line; T displays lines at and below the current line.

To display JCF lines above and below the current line, execute the P (Page) command.

The display depends upon your location in the JCF. The following examples illustrate these variations.

At the Top of the JCF

When you are at the top of the JCF and enter P, CVNC displays the first five JCF lines.

```
NC:> P <----- (P at the JCF top)
.Top.
1   > [JCF CVNC MILL HEADER LINE] <
1   DATUM 'TOP'
1   HOMEPT FROM X -2.1 Y -24 Z 0 ;
1   DEFTOOL '1/4-MILL' MILL7 DIA .25
CORNER 0 HEIGHT 5 GAGE 5
1   DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5
```

When you are at the top of the JCF and enter P3, CVNC displays the first three JCF lines.

```
NC:>P3 <----- (P3 at the JCF top)
.Top.
1   > [JCF CVNC MILL HEADER LINE] <
1   DATUM 'TOP'
1   HOMEPT FROM X -2.1 Y -24 Z 0 ;
```

Within the JCF

When you are within the JCF and enter P, CVNC displays the current line and the two lines preceding and following the current line (five lines in all).

```
NC:>P <----- (P within the JCF)
1   DATUM 'TOP'
1   HOMEPT FROM X -2.1 Y -24 Z 0 ;
1   > DEFTOOL '1/4-MILL' MILL7 DIA .25 CORNER 0 HEIGHT 5 GAGE 5 <
1   DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5
1   PLANE ZWORK -.5
```


When you are within the JCF and enter P3, CVNC displays the current line and the line preceding and following the current line (three lines in all).

```
NC:>P3 <----- (P3 within the JCF)
1     HOMEPT FROM X -2.1 Y -24 Z 0 ;
1   > DEFTOOL '1/4-MILL' MILL7 DIA .25 CORNER 0 HEIGHT 5 GAGE 5 <
1     DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5
```

Within the JCF and with 4 specified, P displays the current line and the two lines preceding and the line following the current line (four lines in all).

```
NC:>P4 <----- (P4 within the JCF)
1     DATUM 'TOP'
1     HOMEPT FROM X -2.1 Y -24 Z 0 ;
1   > DEFTOOL '1/4-MILL' MILL7 DIA .25 CORNER 0 HEIGHT 5 GAGE 5 <
1     DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5
1     RETRACT
1     MOVE HOME
```

When you are within the JCF and enter P:, CVNC displays the current line and the two lines preceding and following the current line (five lines in all) identified by a digitize on the active tool path.

```
NC:>P: MODEL ent d <----- (P followed by a digitize)
1     APPROACH
1     MOVE XYLOC X 0 Y 0 ;
1   > POCKET $L1 $L2 $L3 $L4 ; <
```

Please note: You cannot use this command in the PREPARE mode of CVNC Batch.

To display JCF lines starting at the current line, execute the T (Type) command. For example,

```
NC:>T <----- (T at the JCF top)
.Top.
> [JCF CVNC MILL HEADER LINE ] <
```

With no modifier specified, T displays the current JCF line. In addition, you can enter:

- Tn, replaces n with the number of lines to display.
- TL, displays the next active line; nonexecutable headers and system markers are skipped.
- T:, displays the line identified by a digitize.

Please note: You cannot use this command in the PREPARE mode of CVNC Batch.

Stepping Forward and Backward — F, B, and RESTART

Moving through the JCF

You can move forward and backward in the JCF, controlling the position of the current line pointer with the F, B, and RESTART commands.

As you enter each command line and press RETURN in a new JCF. The current line pointer moves forward and the command you have just entered is processed. You may need to return to an earlier line to change information as you are creating your new JCF.

In this case, use the B (Backward) command to step back in your JCF, undoing (not deleting) the action of each command line you have crossed.

To return to your original position, execute the F (Forward) command, specifying the number of lines to move forward.

To undo the action of all the JCF lines and return to the top, execute the RESTART command.

Inserting New JCF Lines

New lines are inserted above the current line. To insert a new JCF line, move the pointer to the line below the position of the new line and enter the information.

To display a newly inserted line in relation to the other lines, execute the P command after you have inserted the new line. For example,

```
1     PROFILE ON $L1 $L2 $L3 $L4 ;
1     RETRACT
1   > MOVE HOME <
1     [DONE]
.Bottom.
NC:>#> NOW MOVE TO THE HOME POSITION
NC:>P
1     #> NOW MOVE TO THE HOME POSITION
1   > MOVE HOME <
1     [DONE]
.Bottom.
```

Stepping Forward — F

To move the pointer forward a number of lines in the JCF to a new current line, execute the F (Forward) command. For example,

```
NC:>P3
.Top.
1  > [JCF CVNC MILL HEADER LINE ] <
1  DATUM 'TOP'
1  HOMEPT FROM X -2.1 Y -24 Z 0 ;
NC:>F <----- (F steps forward one line)
1  > DATUM 'TOP' <
```

In this example, F moves the pointer forward one line from the current line.

As the pointer moves forward

- All the JCF lines are processed up to the new current line (unless command processing has been turned off with the OFF command).
- All tool path graphics are displayed for motion commands.
- System variables associated with the new line position are set.

F Options

You can enter:

- F, to move the pointer forward one line.
- Fn, replacing n with a specific number of lines to which to move forward.
- F., to move forward to the next file mark (see “Marking and Unmarking Lines — . and -” on page 4-15).
- F/string, to move the pointer forward to the specified string.

Please note: The text delimiter character can be a slash (/), comma (,) or semicolon (;).

Stepping Backward — B

To move the current line pointer backward a number of lines in the JCF to a new current line, execute the B (Backward) command. For example,

```
NC:>P3
.Top.
1  [JCF CVNC MILL HEADER LINE ]
1  DATUM 'TOP'
1  HOMEPT FROM X -2.1 Y -24 Z 0 ; <
```

```
NC:>B2 <----- (B2 steps back two lines)
1  ^  DATUM 'TOP'
1  ^> [JCF CVNC MILL HEADER LINE ] <
.Top.
```

In this example, B2 moves the pointer back two lines from the current line. As the pointer moves backward, it steps over all lines up to and including the new current line. It does not delete them. The pointer also removes tool path graphics associated with each line and resets system variable values to correspond with the new line position.

B Options

You can enter:

- B, to move the pointer back one line.
- Bn, replacing n with a specific number of lines to which to move back.
- B., to move back to the previous file mark (see “Marking and Unmarking Lines — . and -” on page 4-15).
- B/string, to move back to the specified string.

Please note: The text delimiter character can be a slash (/), comma (,) or semicolon (;).

- B:, to move back to the command that generated the specified graphic tool path segment. (For CVNC-P2, B: moves back to the command that generated the NFIG.)

Please note: You cannot use this command in the PREPARE mode of CVNC Batch.

Returning to the JCF Top — RESTART

To move the pointer directly to the first line in the JCF, execute the RESTART command. For example,

```
NC:> P99
.Top.
1  [JCF CVNC MILL HEADER LINE ]
1  DATUM 'TOP'
1  HOMEPT FROM X 5 Y 5 Z 5 ;
1  DISPLAY TOOL DYN VIEW "B" ;
1  POCKET $L1 $L2 $L3 $L4 ;
1  > [DONE] < <----- (Current line)
.Bottom.
```

```
NC:>>RESTART <----- (RESTART executed)
NC:>P99
.Top.
1  > [JCF CVNC MILL HEADER LINE] <
<----- (Current line)
1    DATUM 'TOP
1    HOMEPT FROM X 5 Y 5 Z 5 ;
1    DISPLAY TOOL DYN VIEW "B" ;
1    POCKET $L1 $L2 $L3 $L4 ;
1    [DONE]
.Bottom.
NC:> >RESTART
```

In this example,

1. P99 displays the JCF, where the pointer is positioned at the bottom of the JCF.
2. RESTART is executed.
3. P99 displays the JCF. Now the pointer is positioned at the top of the JCF.

Please note: Any tool path graphics created by the JCF are removed after the RESTART command is executed.

Ending Your JCF Session — EXIT and QUIT

To exit CVNC and save your JCF changes, use the EXIT command. To quit CVNC and discard any changes you have made to your JCF during this session, use the QUIT command.

You can use the EXIT or QUIT command to leave your CVNC session and return to the CADD5 prompt.

EXIT

- Files your JCF and returns you to the CADD5 prompt.
- Overwrites any previously existing JCF with the same name.
- Saves the tool motion, geometry, and entities tagged.
- Creates an output file if OUTPUT was entered to initialize output.

QUIT

- Returns you to the CADD5 prompt.
- Restores the CADD5 database and graphics to its former state.
- Does not file this version of your JCF.

For example,

```
.Top.  
1 > [JCF CVNC MILL HEADER LINE ] <  
1 DATUM 'TOP'  
1 PLANE ZWORK -.5  
1 PLANE ZCLEAR 2  
NC:>EXIT  
#nn# EXIt PART F  
.Top.  
1 [JCF CVNC MILL HEADER LINE]  
1 DATUM 'TOP'  
  
1 PLANE ZWORK -.5  
1 FEED CONNECT 4  
1 > #>THIS VERSION NOT WORTH SAVING <  
NC:>QUIT  
#nn#
```

Please note: Always file your part at the CADD5 prompt. (If you execute EXIT PART QUIT at this point, the CADD5 database does not save the tool motion, geometry, or entities tagged during your CVNC session.)

Controlling the Display — SEE, BR, FU, and GR

As you move forward and backward in the JCF, all lines are displayed on the screen. You can switch between two display modes for tags and location data with SEE. Use the BR command to see only the current line. To restore full output display, use the FU command. You can control the frequency of graphics updates using GR.

Displaying Tags and Location Data — SEE

You can switch between a long and short display mode for tags and location data in JCF command lines. (The normal (default) mode is long.) Enter SEE to switch from long mode to short. Enter SEE again to return to long mode.

Long Mode

In this mode (the default), entity tags and locations are individually displayed in your JCF.

Short Mode

In this mode, only the number of entities or locations appear in the JCF. For example,

Here is a line containing tags in long mode:

```
1 NCGROUP A $T1 $T2 $T3 $T4 $T5 $T6 ;
```

Here is the same line in short mode:

```
1 NCGROUP A <6 ents >;
```

Here is a line containing location data in long mode:

```
1 NCGROUP B X 1.234 Y 2.2348 Z 0 X 2.3434 Y 3.2345 Z 0 ;
```

Here is the same line in short mode:

```
1 NCGROUP B <2 locs >;
```

Please note: When the tag display uses less space than <n ents>, the tags remain fully displayed. Thus, CUT TO ENT \$L1 ; does not become CUT TO ENT <1 ent>;.

Editing with ED or E

These commands display the JCF text in normal (long) mode; that is, locations and tags are visible and in the correct position for editing.

Searching for Strings

When searching for a text string (for example, with F, B, or C) - CVNC always searches for the text as if long mode is in effect.

SEE Examples

```
1   AREAMILL LACE BOUND $L1 $L2 $L3 $L4 A $L5 $L6 $L7 ;
NC:>SEE
NC:>T
   AREAMILL LACE BOUND <4 ents> A <3 ents>;
```

In this example, it outputs the first four tags as <4 ents>; retains the explicit NCGROUP name, outputs the last three tags as <3 ents>.

```
1       CUT XYLOC X 1 Y 1 X 1 Y 2 X 2 Y 2 $P1 $P2 $P3 X 6 Y 5 X 7 Y
5 $P4 $P5;
NC:>SEE
NC:>T
1       CUT XYLOC <3 locs> <3 ents> <2 locs> $P4 $P5
```

In this example, the user entered a mixture of locations and point tags using the .: SEE displays the total number of entities and locations for each mode.

Suppressing JCF Line Printing — BR and FU

To suppress output to the screen as you move forward or backward in your JCF, displaying only the new current line, execute the BR (Brief) command.

Please note: Full display is the default when you enter CVNC.

For example,

```
NC:>P
.Top.
1   > [JCF CVNC MILL HEADER LINE ] <
1   DATUM 'TOP'
1   HOMEPT FROM X -2.1 Y -24 Z 0 ;
1   DEFTOOL '1/4-MILL' MILL7 DIA .25 CORNER 0 HEIGHT 5 GAGE 5
1   DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5
NC:>BR <----- (BR initiates brief output)
F3
```



```
1 > DEFTOOL '1/4-MILL' MILL7 DIA .25 CORNER 0 HEIGHT 5 GAGE 5 <
B3
1 ^ >[JCF CVNC MILL HEADER LINE ] <

.Top.
NC:>FU <----- (FU returns to full output)
NC:>F3
1 DATUM 'TOP'
1 HOMEPT FROM X -2.1 Y -24 Z 0 ;
1 > DEFTOOL '1/4-MILL' MILL7 DIA .25 CORNER 0 HEIGHT 5 GAGE 5 <
NC:>B3
1 ^ HOMEPT FROM X -2.1 Y -24 Z 0 ;
1 ^ DATUM 'TOP'
1 ^ >[JCF CVNC MILL HEADER LINE ] <
.Top.
```

In this example, it executes the BR command first. When it executes the F or B command, it displays the new current line (intermediate lines are suppressed).

After executing the FU (Full) command, it resumes full output. Now when it executes the F or B command, all lines being processed are displayed.

Please note: P and T are not affected by BR.

Viewing Updated Graphics Displays — GR

Each time you step forward over one line in your JCF, you can view the tool path string graphics updated in the CADD5 graphics window to reflect the current state.

If, however, you step forward or backward over a number of lines using a single command, the intermediate graphics displays will, by default, be suppressed until the NC:> prompt reappears. At this time the graphics display will be updated to reflect the final state. This allows relatively fast repositioning within a large JCF.

You can change the mode of operation within CVNC to give more frequent graphics displays by entering GR at the NC:> prompt.

In this mode, if you make a JCF change and then step forward over a number of lines, the graphics displays will update for each line processed. This allows you to see the tool path as it is created.

To revert to less frequent update of the graphics displays, enter GR again.

To display the current graphics display mode, you can view the current state of the #GR system variable associated with GR using the SHOW command. For example,

```
NC:>SHOW #GR
```

The #GR system variable is set to “ON” by default, to indicate that the infrequent graphics display mode is active.

Alternately, it displays “OFF” if the default mode has been switched off.

Please note: You cannot use this command in the PREPARE mode of CVNC Batch.

Managing Changes to Existing JCFs - LOCK, UNLOCK, FC, and RECOVER

If you make changes to an existing JCF, the binary file for that JCF changes. This is because CVNC re-evaluates the changed commands. In addition, the effect of these changes may be carried forward to other commands in the JCF, sometimes leading to unexpected changes in the tool path. Unexpected changes in tool path may also occur due to reasons like enhancements in CVNC, regression or changes in the processor and environment. In such cases you may want to just see the new tool path and revert back to the existing version if the results are not satisfactory.

- The LOCK command allows you to lock the evaluation of commands. CVNC then does not re-evaluate locked commands unless you unlock and edit them. In other words, you can avoid cumulative changes where you do not want them.
- Use the UNLOCK command to unlock locked commands and enable their evaluation.
- You can use the FC command to compare the existing and new evaluations and display the differences in the form of text or graphics.
- The COMPARE command allows you to set options for the display of the above differences.
- Lastly, use RECOVER to revert back to the original evaluation.

Locking a Command - LOCK

Use the LOCK command to lock the evaluation of commands in the JCF. CVNC does not re-evaluate locked commands unless you explicitly unlock and edit them. For example,

```
NC:>T
1      > PROFILE TO $L1 $L2 ; <
NC:>LOCK
NC:>T
.locked.> PROFILE TO $L1 $L2 ; <
```

When you lock a command, CVNC marks it as shown in the example above. You can even lock a command that has been withdrawn, to retain its evaluation.

Locking a command ensures that its evaluation does not change. It also requires that the starting point for the tool path generated by the command does not change. Such a situation may cause an error if the previous command does not leave the tool at the starting point of the locked command. CVNC stores the coordinates of this point in #FRSTLOC (see the *CVNC System Variables Guide* for details of this system variable.).

LOCK Options

You can enter:

- LOCK, to lock the current line in the JCF.
- LOCKn, to lock n lines in the JCF from the current line, replacing n with the specified number.
- LOCK/string, to lock the JCF from the current line up to the specified string.

Please note: Locked commands do not respond to changes in geometry that they refer to. Locking also disables the parametric update for the locked command (subsequent unlocking enables it again). When you lock a command, its text may not match the tool path evaluation in the binary file.

Unlocking a Command - UNLOCK

Use the UNLOCK command to unlock a locked command. When you unlock a command, CVNC re-evaluates it if its previous evaluation has been affected by any changes. It is re-evaluated when you move forward over it.

CVNC removes lock marks when you unlock a command, as shown in the following example:

```
NC:>T  
.locked.> PROFILE TO $L1 $L2 ; <  
NC:>UNLOCK  
NC:>T  
1      > PROFILE TO $L1 $L2 ; <
```

UNLOCK Options

You can enter:

- UNLOCK, to unlock the current line in the JCF.
- UNLOCKn, to unlock n lines in the JCF from the current line, replacing n with the specified number.
- UNLOCK/string, to unlock the JCF from the current line up to the specified string.

Comparing Evaluations - FC

Use the FC (Forward Compare) command to compare the new and old evaluations and display differences in the form of text or graphics.

If there are no differences, CVNC displays the following message:

```
No Differences Found.
```

You can see the list of differences, if any, as shown in the following example:

```
At Toolpath Point: 1
  [1 Old] x -4.849391e+00 Y -2.027654e+00 Z 0.000000e+00
  [1 New] X -5.322951e+00 Y -1.867213e+00 Z 0.000000e+00
At Toolpath Point: 2
  [2 Old] X -3.236564e+00 Y 2.732787 Z 0.000000e+00
  [2 New] X -3.595082e+00 Y 3.232787 Z 0.000000e+00
At Toolpath Point: 3
  [3 Old] X -1.689376e+00 Y 2.732787 Z 0.000000e+00
  [3 New] X -1.783607e+00 Y 3.232787 Z 0.000000e+00
```

FC Options

You can enter:

- FC, to compare the current line in the JCF.
- FCn, to compare n lines in the JCF from the current line, replacing n with the specified number.
- FC., to compare the JCF from the current line up to the next file mark (see “Marking and Unmarking Lines — . and -” on page 4-15).
- FC/string, to compare the JCF from the current line up to the specified string.

Please note: The text delimiter character can be a slash (/), comma (,) or semicolon (;).

The COMPARE Command

You can use this command to specify whether the differences between the old and new evaluations should be displayed on the screen (graphic and non-graphic data) or saved to a file (non-graphic data only). This command also allows you to set the tolerance value for the differences and the color in which these differences should be displayed on the screen.

Please note: This command is a NOJCF command and applies to the whole JCF.

The following examples illustrate the use of the COMPARE command.

```
NC:> COMPARE filename TOLER nexp
```

allows you to specify a text file to save the differences. The TOLER modifier specifies the tolerance.

```
NC:> COMPARE SCREEN COLOR RED TOLER nexp
```

specifies that the differences should be displayed on the screen in the specified color. SCREEN is mutually exclusive with filename. The colors can be BLUE, GREEN or RED.

```
NC:> COMPARE SCREEN COLOR OFF TOLER nexp
```

allows you to set the display of differences to off.

Recovering the Original Evaluation - RECOVER

The RECOVER command allows you to recover the earlier evaluation of commands (the one existing when FC was used). You must position the pointer to the desired commands before using RECOVER, and then move forward over them.

```
NC:>RECOVER
```

RECOVER also locks the command.

Saving JCF Contents — SECURE, SAVE, and RESTORE

SECURE Procedure

SECURE allows you to save your JCF contents and CADDs part during a work session. After you enter the SECURE command, the `NC : >` prompt returns, so you can continue your work.

Please note: Periodically executing the SECURE command during a JCF creation or editing session minimizes the possible effect of any unexpected system failures.

The following files are overwritten or created by SECURE:

- `_jcf` binary file
- `_sym` symbol table file
- `_pd` part data file
- `_cmp` file
- JCF text file
- NCpunch configuration file (CVNC-P2 only)
- Rougher range map file (CVNC-M3 and CVNC-M5 only)

For example,

```
NC:>homept from x 0 y 0 z 10;  
NC:>Display off  
NC:>cut to $L1;  
NC:>DISPLAY TOOL CONST 1 PAINT  
NC:>CUT TO $L2;  
NC:>CUT YINC 0.5  
NC:>CUT TO $L3;  
NC:>ECHO COLOR RED $L3 $L2;  
NC:>SECURE  
NC:>
```

In this example, after SECURE is executed,

1. Tool path graphics from the home point to \$L1 briefly become visible.
2. The red color for the entities \$L2 and \$L3 temporarily disappears.
3. The tool at the current location briefly disappears.
4. The system pauses while the part is filed.

5. Graphics for the tool path from the home point to \$L1 disappear.
6. The red color for the entities \$L2 and \$L3 returns.
7. The tool is redisplayed at its current location.

Please note: Temporary graphics for the painted tool path remain visible, but are not preserved in the saved part.

Here is another example:

```
CVNC MILLING
Entering NEW JCF  'BRACKET.JCF.11'
NC:>P
.Top.
1      [JCF  CVNC MILL HEADER LINE ]
1    > [DONE] <
.Bottom
NC:>DATUM 'TOP'
CPLANE ACTIVATED
Setting tool location system variables:
#CURLOC is now X 0.0 Y 0.0 Z 0.0  in active    cplane space.
NC:>HOMEPT FROM X 2 Y 2 Z 0;
NC:>SECURE <----- (SECURE is entered now)
NC:>#> A SYSTEM FAILURE IS ABOUT TO OCCUR <----- (This line will
be lost)
```

(A system failure occurs.)

```
%cadds
#nn#ACTivate PART BRACKET
Activating OLD part.
#nn#ACTivate DRAW 1
#nn#PROGRAM NCMill OPR 11 <----- (Same JCF name)
CVNC MILLING
Entering OLD JCF  'BRACKET.JCF.11'
NC:>P
.Top.
1      [JCF  CVNC MILL HEADER LINE ]
1    > DATUM 'TOP' <
1      HOMEPT FROM X 2 Y 2 Z 0;
1      [DONE]
.Bottom. <----- (Original contents are now restored to point
where SECURE was entered, but comment line entered afterwards is
lost)
```

In this example,

1. The user starts a new JCF and enters the DATUM and HOMEPT commands.
2. SECURE saves the contents of the JCF and part at this point.

3. A system failure occurs.
4. The JCF is reentered. Note that you are positioned at the point when SECURE was executed.
5. P (Page) displays the JCF contents. Note that the comment line was lost.

Please note: You cannot use this command in the PREPARE mode of CVNC Batch.

SAVE Procedure

The SAVE command saves the text JCF under the `jcfsave` directory. The full directory path depends on the JCF name type. For example,

Table 3-1 Directory Paths for Text JCFs

JCF Name Type	SAVE Directory
filename	<code>jcfsave/_bcd/filename</code>
EXT	<code>jcfsave/partname/_bcd/extension</code>
OPR	<code>jcfsave/partname/jcf/_bcd/operation_number</code>

To save your JCF contents, execute SAVE. For example,

```
NC:>SAVE
```

Please note: Only the text JCF version is saved; the JCF must be reexecuted when restored.

RESTORE Procedure

The RESTORE command replaces a current JCF with an existing JCF saved with SAVE.

RESTORE deletes the current JCF and replaces it with the contents of the saved JCF file. The current JCF that is deleted and replaced cannot be retrieved.

To restore your JCF contents, execute RESTORE. For example,

```
NC: >RESTORE
```

Please note: The JCF must be reexecuted, as only a text version of it was saved.

Controlling Processing — SYN, ON, OFF, and EX

To control automatic syntax checking and command processing, use the OFF, SYN, and ON commands. You can temporarily turn execution of JCF lines on and off with the EX command.

CVNC Command Processing and Syntax Checking

When you enter your JCF, the default CVNC setting

- Checks each JCF line for correct syntax and halts processing if the entry contains syntax errors.
- Processes each JCF line as you move forward through the JCF.

Turning Command Processing/Syntax Checking Off — OFF

To turn command processing and automatic syntax checking off, execute the OFF command. This

- Prevents JCF lines from being checked for correct syntax.
- Surpasses processing as you move through the JCF.

Please note: You cannot use this command in the PREPARE mode of CVNC Batch.

Turning Syntax Checking On/Processing Off — SYN

To turn automatic syntax checking on and command processing off, execute the SYN command. This

- Allows each JCF line to be checked for correct syntax.
- Surpasses processing as you move through the JCF.

Please note: You cannot use this command in the PREPARE mode of CVNC Batch.

Turning Command Processing/Syntax Checking On — ON

To turn automatic syntax checking and command processing back on, execute the ON command.

Please note: You cannot use this command in the PREPARE mode of CVNC Batch.

Turning Command Execution On and OFF — EX

You can use EX to temporarily turn execution of the current line off and on. The first time you enter EX, current line execution is turned off. Enter EX again to turn execution back on.

Lines marked for nonexecution are output in the JCF text file surrounded by parentheses (). This means you can delete the binary files associated with a JCF and still have lines marked for nonexecution.

You can also manually edit lines for nonexecution offline (for example, with the UNIX vi command) by placing parentheses () at the beginning and end of the lines.

Please note: When a command that produces graphical output is marked for nonexecution, stepping backward and forward over this line does not result in the display of its associated graphics.

For commands continued onto subsequent lines, you need only include the parentheses () around the first line. The reverse case is also true. You can turn on again a line that appears in the JCF text file marked for nonexecution by removing the parentheses from the first line — there is no need to edit subsequent continuation lines.

Please note: You cannot execute EX for lines currently marked as inactive with square brackets [], such as decomposed command lines and the header and DONE lines. Also, you cannot use any other CVNC editor command (for example, E) to insert the nonexecution brackets or use EX within star buffer mode.

For example,

```
NC:>P10 <----- (P10 displays the JCF)
.Top.
1          [JCF  CVNC MILL HEADER LINE ]
1          DATUM 'TOP'
1          PLANE ZRETRACT .6
1          > PLANE ZWORK -0.01 <
1          [DONE]
.Bottom.
NC:>EX <----- (First time EX is entered)
NC:>P10
.Top.
1          [JCF  CVNC MILL HEADER LINE ]
```

```
1          DATUM 'TOP'
1          PLANE ZRETRACT .6
1          > (PLANE ZWORK -0.01) < <-----
1          [DONE] <----- (Line marked for nonexecution)
.Bottom.
NC:>EX <----- (EX is entered again)
NC:>P10

.Top.
1          [JCF CVNC MILL HEADER LINE ]
1          DATUM 'TOP'
1          PLANE ZRETRACT .6
1          > PLANE ZWORK -0.01 < <-----
1          [DONE] <----- (Line restored for execution)
.Bottom.
```

In this example,

1. P10 displays the JCF. The current line contains the PLANE ZWORK command.
2. Enter EX. Rounded brackets now mark this line for nonexecution.
3. Enter EX again. The rounded brackets are removed and the command is restored for execution.

Packing the JCF — PACK and PACK JCF

During part and JCF editing, the system leaves extra space in the database due to expansion and contraction of the data. The CVNC and CADD5 PACK commands reclaim this space, thereby minimizing storage disk use and maximizing database access speed.

CVNC PACK Procedure

To remove wasted space from the binary JCF files (`jcfname._jcf` and `jcfname._cmp`), execute the PACK command. For example,

```
#nn#PROG NCMill OPR 2
CVNC MILLING
Entering OLD JCF 'BRACKET.JCF.2'
NC:>PACK
Packing JCF ... Please wait.
```

You can enter PACK anywhere in the JCF. It does not become a permanent part of the JCF.

Please note: The pack process is a time-consuming function. It may be appropriate to perform PACK during off hours. You cannot use this command in the PREPARE mode of CVNC Batch.

CADD5 PACK Procedure

You can also use the CADD5 PACK JCF command to pack all JCFs associated with the active part. It processes only JCFs that can or need to be packed. Therefore if a JCF needs to be reexecuted because the tool paths have been deleted, it is not to be packed. PACK JCF also tests the JCF to see if they have been modified since they were last packed.

For example,

```
#nn#PACK JCF
***** Packing JCFs for this part *****
Packing JCF PART.AMILL1.JCF.A3! ...
Packing JCF PART.AMILL1.JCF.A5! ...
Packing JCF PART.AMILL1.JCF.A9! ...
Packing JCF PART.AMILL1.JCF.11! ... already packed.
Packing JCF AREAMILL.JCF.19! ... There are no binary files to
pack.
*****      Packing JCFs done      *****
```

In this example,

1. Three JCFs are packed.
2. One was already packed.
3. One could not be packed because the binary files had been deleted.

ESTIMATE Modifier: The PACK JCF ESTIMATE command scans the JCFs to determine the potential space savings that could be achieved if you execute the PACK JCF command. It informs you of the estimate and prompted to affirm or cancel the command.

For example,

```
#nn#PACK JCF Estimate
Text Window

#4#PACK JCF Estimate
JCF opr  %saving  Bytes saving
-----  -
   3   97.8    621221
   6   98.4    811546
   7   40.7    2117155
  11  PACKED
  19  - No binary files present.
TYPE "OK" TO PACK THE JCFs
```

In this example, by using the ESTIMATE modifier you can determine the amount of space that will be saved; then the system gives you the option to pack the JCFs.

Please note: The CADDs PACK JCF command is not available while in RESU mode.

This chapter shows you how to use editing commands to change information in your JCF.

- Editing Commands Overview
- Editing a Line — E
- Editing Graphic Entities — ED
- Substituting Text — C
- Creating Comment Lines — #>
- Appending Text — +
- Deleting Lines — D
- Canceling a Deletion — UNDEL
- Repeating Command Lines — A and !!
- Marking and Unmarking Lines — . and -
- Selecting and Copying Lines — # and =
- Cutting and Pasting Lines — < and >

Editing Commands Overview

Editing JCF Lines

In a new JCF, as you enter each command line and press RETURN, the current line pointer moves forward and the command just entered is processed.

These command lines can be edited in a number of ways. For example, you can

- Edit the line using standard editing characters.
- Edit an entity by selecting its corresponding tool path string.
- Substitute, prefix, replace, or delete text information.

Use the following commands to make these changes:

- E (Edit)
- ED (Edit Digitize)
- C (Change)
- + (Append)
- D (Delete)
- UNDEL

Deleting Line Information Before it is Processed

To correct typing errors as you initially enter a JCF line, press the DELETE key back to the incorrect character and retype the rest of the line.

To delete the last word, press CONTROL-W.

To delete the entire line and begin again, press CONTROL-E.

Copying and Moving JCF Lines

JCF lines can be:

- Copied
- Moved

Using the commands described in this chapter, you can:

- Repeat the same application command line.
- Copy a single JCF line to another location.
- Copy multiple consecutive JCF lines to another location.
- Cut a line from one location in your JCF and move it to another location.
- Cut a series of lines from one JCF location and move them to another location.

For example, using a combination of the F (Forward) and !! (Repeat last application command) commands, you can easily and quickly insert all the SPEED ON statements in your JCF.

Editing a Line — E

To edit the current JCF line, execute the E command. After you enter E, the current line is displayed with the cursor under the first character. For example,

```
NC:> E
PLANE ZCLEAR 2
NC:>
```

To perform the following actions, use the specified keys.

Key	Action
SPACE BAR	Moves the cursor to the RIGHT one space in the line without deleting any of the characters in the line.
DELETE	Moves the cursor to the LEFT one space in the line without deleting any characters in the line.
^	Replaces the character at the current cursor location with a space.
\	Deletes a character or space at the current cursor location.
[, followed by the character(s) to be inserted	Inserts one or more characters or spaces at the current cursor locations
Enter character(s)	Substitutes or adds one or more characters at the current cursor location.
RETURN	Continues editing the line and sees the results of the last edit action.
RETURN	Saves the edited line and returns to the NC:> prompt.

Editing Example

```
1          > PLANE ZCLEAR 2 <
NC:> E
          PLANE ZCLEAR 2
                5
          PLANE ZCLEAR 25
                \
          PLANE ZCLEAR 2
                ^
          PLANE ZCL AR 2
                [E
          PLANE ZCLE AR 2
                \
          PLANE ZCLEAR 2
```

NC : >

In this example, various edits are made to the PLANE ZCLEAR line using E:

1. Add a character (5) to the line.
2. Delete the 5 by using the backslash (\).
3. Insert a blank space is using the caret (^).
4. Insert a letter with [E.
5. Delete a blank space.

The RETURN key is pressed twice at the end of the editing session to return to the NC : > prompt.

Please note: If the information contained in the current line cannot fit on a single line and continues onto the next line (or for several more lines), use the format En to specify which line to edit.

Editing Graphic Entities — ED

The ED command allows you to redigitize a single entity or a group of entities.

After you execute ED, the cursor moves to the first selected entity (or group of entities).

To make a correction, enter a colon (:) and digitize the correct entity.

If the entity is correct, press TAB to reinsert the correct entity and position the cursor at the next digitized entity.

To end the edit digitize session, press RETURN.

For example,

```
NC:> P1
1 > CUT TO $L102 ; INTOF TO $L101 ;BIAS X -1 Y 0 ; <
NC:> ED
CUT TO $L102 : MODEL ent d
CUT TO $L103 ;
          INTOF TO $L101 ;
          INTOF TO $L101 ;BIAS X -1 Y 0 : MODEL loc X1Y2
CUT TO $L103 ; INTOF TO $L101 ;BIAS X 1 Y 2 ;
NC:> P1
1 > CUT TO $L103 ; INTOF TO $L101 ;BIAS X 1 Y 2 ; <
NC:>
```

In this example,

1. P1 displays the JCF line.
2. ED is executed. CUT TO \$L102 ; appears on the screen.
3. A colon (:) is entered. CVNC prompts MODEL ent .
4. A new line (\$L103) is digitized. CVNC displays the new CUT TO \$L103 and on the next line displays INTOF to \$L101 ;.
5. The TAB is pressed because this digitize is correct. The system displays INTOF TO \$L101 ; BIAS X -1 Y 0.
6. A colon (:) is entered and CVNC prompts MODEL loc.
7. X1Y2 is entered. The ED session is finished. The edited line is displayed and the NC : > prompt returns.
8. P1 shows the final edited JCF line.

Substituting Text — C

Replacing Text

To replace a specified string of text in the current line with a new string, execute the `C/string/string` command.

Prefixing Text

To prefix the current line with a specified string, execute the `C//string` command.

Deleting Text

To delete a specified string in the current line, execute the `C/string//` command.

Please note: The text delimiter character can be a slash (/), comma (,) or semicolon (;).

C (Change) Procedures

The following example illustrates various C procedures:

```
NC:> P99
.Top.
1 [JCF CVNC MILL HEADER LINE ]
1 DATUM 'TOP'
1 HOMEPT FROM X 0 Y -2 Z 5;
1 DEFTOOL '1/4-MILL' MILL7 DIA .25
CORNER 0 HEIGHT 5 GAGE 5
1 CHGTOOL 1 '1/4-MILL'
1 > PROFILE TO $L3 $L2 $L4 $L1 ;ROLL <
1 [DONE]
.Bottom.
NC:> C/TO/ON

1 > PROFILE ON $L3 $L2 $L4 $L1 ;ROLL <
NC:> B/DEFTOOL
1 CHGTOOL 1 '1/4-MILL'
1 ^> DEFTOOL '1/4-MILL' MILL7 DIA .25
CORNER 0 HEIGHT 5 GAGE 5
NC:> C/#>
1 > #> DEFTOOL '1/4-MILL' MILL7 DIA .25
CORNER 0 HEIGHT 5 GAGE 5<
NC:> F/PROFILE
1 CHGTOOL 1 '1/4-MILL'
1 > PROFILE ON $L3 $L2 $L4 $L1 ;ROLL <
NC:> C/ROLL//
1 > PROFILE ON $L3 $L2 $L4 $L1 ; <
```

```
NC:> P99
.Top.
1      [JCF  CVNC MILL  HEADER LINE ]
1      DATUM 'TOP'
1      HOMEPT FROM X 0 Y -2 Z 5;
1      #> DEFTOOL '1/4-MILL' MILL7 DIA .25
CORNER 0 HEIGHT 5 GAGE 5
1      CHGTOOL 1 '1/4-MILL'
1      > PROFILE TO $L3 $L2 $L4 $L1 ; <
1      [DONE]
.Bottom.
NC:>
```

In this example,

1. P99 displays the JCF. The current line contains the statement PROFILE TO \$L3 \$L2 \$L4 \$L1 ;ROLL.
2. C/TO/ON edits the line to PROFILE ON \$L3 \$L2 \$L4 \$L1 ;ROLL.
3. B/DEFTOOL moves the pointer back to the DEFTOOL '1/4-MILL' MILL7 . . . line.
4. C//#> edits the line to #> DEFTOOL '1/4-MILL' MILL7. (It is now a comment line.)
5. F/PROFILE moves the pointer forward to the PROFILE ON \$L3 \$L2 \$L4 \$L1 ;ROLL line.
6. C/ROLL// edits the line to PROFILE ON \$L3 \$L2 \$L4 \$L1 ;.
7. P99 displays the edited JCF.

Creating Comment Lines — #>

To add a comment line (a nonexecuted line) to your JCF, enter a hash mark (#) followed by a right angle bracket (>) in columns 1 and 2. For example,

```
.Top.
1      [JCF CVNC MILL HEADER LINE]
1      DATUM 'TOP'
1      HOMEPT FROM X -2.1 Y -24 Z 0 ;
1      DEFTOOL '1/4-MILL' MILL7 DIA .25
CORNER 0 HEIGHT 5 GAGE 5
1      DEFTOOL 'DRILL' DRILL DIA .25 GAGE 5
1      > PLANE ZWORK -.5 <
1      PLANE ZCLEAR 2
NC:> #>
NC:> #>THE Z PLANES ARE SPECIFIED NEXT
NC:> #>
NC:> P6
1      #>
1      #>THE Z PLANES ARE SPECIFIED NEXT
1      #>
1      > PLANE ZWORK -.5 <
1      PLANE ZCLEAR 2
```

In this example,

1. The user adds a blank comment line to the JCF.
2. The user adds a comment.
3. The user adds another blank comment line(improves readability).

Please note: Comments can contain a maximum of 70 characters (including blank spaces).

Appending Text — +

To append text to a JCF line, execute the + (Append) command. For example,

```
1 > #>THE CONSTRUCTION PLANES ARE SPECIFIED <  
NC:> +/ NEXT  
1 > #>THE CONSTRUCTION PLANES ARE SPECIFIED NEXT <
```

In this example, the string NEXT is appended to the current line.

Please note: The leading text delimiter character can be a slash (/), comma (,), or semicolon (;).

Deleting Lines — D

To delete a JCF line,

1. Move the pointer to the line.
2. Execute the D command.

This action

- Flags the line for deletion.
- Makes the next line the current line after you execute any command except P (Page) or T (Type).

For example,

```
1 > DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5 <
NC:> D
NC:> P
1 DEFTOOL '1/4-MILL' MILL7 DIA .25
CORNER 0 HEIGHT 5 GAGE 5
.deleted DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5 <
1 > PLANE ZWORK -.5 <
```

With no modifiers specified, D deletes the current line. The P (Page) command displays the deleted line. The new current line is the line following the deleted line.

Please note: To immediately reverse the delete action, use the UNDEL command described in the next section. However, entering a command (other than P or T) completes the delete action and UNDEL cannot be used to reverse the delete.

D Options

1. To delete a series of consecutive forward lines, specify Dn, replacing n with the number of lines to be deleted.
2. Specify D. to delete back to the preceding file mark (Chapter 4, “Marking and Unmarking Lines — . and -”, for details on Marking and Unmarking a Line).
3. Specify D/string, to delete forward up to (but not including) string.

Canceling a Deletion — UNDEL

To restore a line flagged for deletion by the preceding D command, execute the UNDEL command. For example,

```
.Top.  
1      [JCF CVNC MILL HEADER LINE]  
.deleted DATUM 'TOP'  
1  >  HOMEPT FROM X -2.1 Y -24 Z 0 ; <  
1      DEFTOOL '1/4-MILL' MILL7 DIA .25  
CORNER 0 HEIGHT 5 GAGE 5  
1      DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5  
NC:> UNDEL  
NC:> P  
.Top.  
1      [JCF CVNC MILL HEADER LINE]  
1  >  DATUM 'TOP' <  
1      HOMEPT FROM X -2.1 Y -24 Z 0 ;  
1      DEFTOOL '1/4-MILL' MILL7 DIA .25  
CORNER 0 HEIGHT 5 GAGE 5  
1      DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5
```

In this example, the DATUM command line was previously deleted.

1. UNDEL restores the line.
2. The new current line is the DATUM line.

Please note: UNDEL cannot restore a deleted line if any command other than P (Page) or T (Type) has been executed following the D (Delete) command.

Repeating Command Lines — A and !!

Repeating Editor Commands — A

To repeat a CVNC editor command, execute the A (Again) command. For example,

```
.Top.
1 > [JCF CVNC MILL HEADER LINE] <
1 DATUM 'TOP'
1 HOMEPT FROM X -2.1 Y -24 Z 0 ;
1 DEFTOOL '1/4-MILL' MILL7 DIA .25
CORNER 0 HEIGHT 5 GAGE 5
1 DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5
NC:> F2
1 DATUM 'TOP'
1 > HOMEPT FROM X -2.1 Y -24 Z 0 ; <
NC:> A
1 DEFTOOL '1/4-MILL' MILL7 DIA .25
CORNER 0 HEIGHT 5 GAGE 5
1 > DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5 <
NC:>
```

In this example,

1. F2 moves the pointer forward two lines.
2. A executes the F2 command again, moving the pointer forward another two lines.

Please note: Use the A command to avoid lengthy typing of complex editor commands.

To repeat the command you have just entered in your JCF (and define the repeated line as the new current line), execute the !! command. For example,

```
NC:> PLANE ZWORK 2
NC:> !!
1 PLANE ZWORK 2
NC:> E
PLANE ZWORK 2
4
PLANE ZWORK 4
NC:> P99
.Top.
1 [JCF CVNC MILL HEADER LINE]
1 PLANE ZWORK 4
```

In this example,

1. The PLANE ZWORK command is executed.
2. The !! command is executed, repeating the PLANE ZWORK command.
3. The repeated line is edited with the E (Edit) command and becomes the new current line, as shown with the P (Page) command.

Please note: To use !!, the last command entered must be an application command line inserted in the JCF.

Marking and Unmarking Lines — . and -

Marking a Line

To place a file mark on the current line, execute the . (Mark) command. For example,

```
.Top.
1      [JCF CVNC MILL HEADER LINE]
1      DATUM 'TOP'
1      > HOMEPT FROM X -2.1 Y -24 Z 0 ; <
NC:> .
NC:> P
.Top.
1      [JCF CVNC MILL HEADER LINE]
1      DATUM 'TOP'
1      . > HOMEPT FROM X -2.1 Y -24 Z 0 ; <
```

In this example, the HOMEPT line is marked. P displays the file mark.

Removing a Mark

To remove a file mark from the current line, execute the - (Delete Mark) command. For example,

```
NC:> P3
1      PLANE ZWORK
1      . > APPROACH <
1      MOVE XYLOC X 0 Y 0 ;
NC:> -
NC:> P3
1      PLANE ZWORK 2
1      > APPROACH <
1      MOVE XYLOC X 0 Y 0 ;
```

In this example,

1. P displays the JCF portion, where a file mark currently exists on the APPROACH command line.
2. The - (Delete Mark) command is executed.
3. P shows that the file mark has been removed.

Selecting and Copying Lines — # and =

To copy a JCF line(s) to another location in your JCF,

1. Mark the line to be selected or the initial line in a series of lines to be selected with the . (Mark) command.
2. Move the pointer forward to the line below the marked line(s).
3. Execute the # (Select) command. Selected lines are indicated with the percent sign (%).
4. Move to the line below the location where the lines are to be copied.
5. Execute the = (Copy) command.

For example,

```
NC:> P99
.Top.
1          [JCF  CVNC MILL HEADER LINE]
1          DATUM 'TOP'
1          HOMEPT FROM X 0 Y -2 Z 5;
1          PLANE ZAPPR 4
1          PLANE ZRETRACT 4
1          PLANE ZWORK 2
1          APPROACH
1          MOVE XYLOC X 0 Y 0 ;
1          POCKET $L1 $L2 $L3 $L4 ;
1          RETRACT
1          MOVE HOME
1          PLANE ZWORK 1
1          > [DONE] <
.Bottom.
NC:> B/APPROACH
1  ^ PLANE ZWORK 1
1  ^ MOVE HOME
1  ^ RETRACT
1  ^ POCKET $L1 $L2 $L3 $L4 ;
1  ^ MOVE XYLOC X 0 Y 0 ;
1  ^> APPROACH <
    ^-----^
NC:> .
NC:> P
1  PLANE ZRETRACT 4
1  PLANE ZWORK 2
1  . >APPROACH <
1  MOVE XYLOC X 0 Y 0 ;
1  POCKET $L1 $L2 $L3 $L4 ;
NC:> F/ZWORK
1  MOVE XYLOC X 0 Y 0 ;
1  POCKET $L1 $L2 $L3 $L4 ;
1  RETRACT
```



```

1      MOVE HOME
1    > PLANE ZWORK 1 <
1      ^---^
NC:> #
NC:> P13
1      PLANE ZWORK 2
1 %    APPROACH
1 %    MOVE XYLOC X 0 Y 0 ;
1 %    POCKET $L1 $L2 $L3 $L4 ;
1 %    RETRACT
1 %    MOVE HOME
1    > PLANE ZWORK 1 <
1 [DONE]
.Bottom.
NC:> F
1    > [DONE] <
.Bottom.
NC:> =
1      APPROACH
1      MOVE XYLOC X 0 Y 0 ;
1      POCKET $L1 $L2 $L3 $L4 ;
1      RETRACT
1      MOVE HOME
1    > [DONE] <
.Bottom
NC:> P19
1      POCKET $L1 $L2 $L3 $L4 ;
1      RETRACT
1      MOVE HOME
1      PLANE ZWORK 1
1      APPROACH
1      MOVE XYLOC X 0 Y 0 ;
1      POCKET $L1 $L2 $L3 $L4 ;
1      RETRACT
1      MOVE HOME
1    > [DONE]
.Bottom.

```

In this example,

1. The P (Page) command displays the JCF.
2. B/APPROACH moves the pointer back to the previous APPROACH line.
3. The APPROACH line is marked with the . (Mark) command.
4. P displays the file mark symbol.
5. F/ZWORK moves the pointer forward to the next ZWORK entry.
6. The # (Select) command is executed.
7. P displays the selected lines preceded with the percent sign (%).
8. F moves the pointer forward.

- 9.** The = (Copy) command is executed.
- 10.** P displays the copied JCF lines in their new position.

Cutting and Pasting Lines — < and >

To move a JCF line(s) to another location in your JCF,

1. Mark the line to be cut or the initial line in a series of lines to be cut with the . (Mark) command.
2. Move the pointer forward to the line below the line(s) to be cut.
3. Execute the < (Cut) command. Lines to be cut are indicated in your JCF with the percent sign (%).
4. Move the pointer to the line below the location in the JCF where the selected lines are to be reinserted.
5. Execute the > (Paste) command.

For example,

```
NC:> P15
1 [JCF CVNC MILL HEADER LINE]
1 DATUM 'TOP'
1 HOMEPT FROM X 0 Y -2 Z 5;
1 > DEFTOOL '1/2 MILL' MILL7 DIA .25
CORNER 0 HEIGHT 5 GAGE 5
1 DEFTOOL '1 MILL' MILL7 DIA .5
CORNER 0 HEIGHT 5 GAGE 5
1 #> TOOL DEFINITION
1 #> CONSTRUCTION PLANE DEFINITIONS
1 PLANE ZAPPR 4
1 PLANE ZRETRACT 4
1 [DONE]
.Bottom.
NC:> .
NC:> P
1 DATUM 'TOP'
1 HOMEPT FROM X 0 Y -2 Z 5;
1 . >DEFTOOL '1/2 MILL' MILL7 RAD .25
CORNER 0 HEIGHT 5 GAGE 5<
1 DEFTOOL '1 MILL' MILL7 RAD .5
CORNER 0 HEIGHT 5 GAGE 5
1 #> TOOL DEFINITION
NC:> F2
1 DEFTOOL '1 MILL' MILL7 RAD .5
CORNER 0 HEIGHT 5 GAGE 5
1 > #> TOOL DEFINITION <
NC:> <
1 > #> TOOL DEFINITION <
NC:> P
1 % DEFTOOL '1/2 MILL' MILL7 RAD .25
CORNER 0 HEIGHT 5 GAGE 5<
1 % DEFTOOL '1 MILL' MILL7 RAD .5
CORNER 0 HEIGHT 5 GAGE 5
```

```
1 > #> TOOL DEFINITION <
1 PLANE ZAPPR 4
1 PLANE ZRETRACT 4
NC:> F
1 > #> CONSTRUCTION PLANE DEFINITIONS <
NC:> >
1 DEFTOOL '1/2 MILL' MILL7 RAD .25
CORNER 0 HEIGHT 5 GAGE 5
1 DEFTOOL '1 MILL' MILL7 RAD .50
CORNER 0 HEIGHT 5 GAGE 5
NC:> P15
1 [JCF CVNC MILL HEADER LINE]
1 DATUM 'TOP'
1 HOMEPT FROM X 0 Y -2 Z 5;
1 #> TOOL DEFINITION
1 DEFTOOL '1/2 MILL' MILL7 RAD .25
CORNER 0 HEIGHT 5 GAGE 5
1 DEFTOOL '1 MILL' MILL7 RAD .5
CORNER 0 HEIGHT 5 GAGE 5
1 > #> CONSTRUCTION PLANE DEFINITIONS <
1 PLANE ZAPPR 4
1 PLANE ZRETRACT 4
1 [DONE]
.Bottom.
NC:>
```

In this example,

1. The P (Page) command displays the JCF.
2. The . (Mark) command places a file mark on the current line.
3. P displays the file mark.
4. F2 moves the pointer forward two lines to the tool definition comment line.
5. The < (Cut) command is executed.
6. P displays the two lines selected to be moved.
7. F moves the pointer forward one line to the construction plane definition comment line.
8. The > (Paste) command is executed.
9. P15 displays the two lines reinserted above the current line.

Please note: You cannot use the < (Cut) command in the PREPARE mode of CVNC Batch.

Accessing CADD5 from CVNC

This chapter describes how to access single Explicit CADD5 commands from within CVNC using the ~ (tilde character) or the IC command, and how to move to the Explicit CADD5 environment using the RESU command (for entering multiple CADD5 commands). The chapter also tells you how to return to CVNC using the RESUME NCPROC command from CADD5.

- Entering Single CADD5 Commands — ~ and IC
- Entering Multiple CADD5 Commands — RESU

Entering Single CADDs Commands — ~ and IC

Accessing CADDs — ~ and IC

You can use various methods to access CADDs to enter a single CADDs command.

Use the ~ (tilde character) to execute any CADDs command available in CVNC and access getdata options.

Alternately, use the IC command to execute a single CADDs command, but IC does not allow getdata options.

(You can also use the RESU command (described in the next module) to execute a single command, but typically you would choose RESU to execute more than one CADDs command.)

Please note: CVNC prevents you from executing CADDs commands that might create inconsistencies between the CADDs and JCF databases.

Moving to CADDs with the ~

To execute a single Explicit CADDs command from CVNC, enter ~, followed by the selected CADDs command, modifiers, and any required getdata information.

After you enter the ~, the prompt changes to NC\$>, indicating that CVNC is ready to receive CADDs information. Execute the CADDs command. For example,

```
NC:> ~LIST LAYER
NC$> LIST LAYer
ACTIVE CONSTRUCTION LAYER: 0
VISIBLE LAYERS:
DRAWING: 1 0-254

VIEW: TOP 0-254
VIEW: FRONT 0-254
VIEW: ISO 0-254

CPLANE ACTIVATED
NC:>
```

The prompt reverts to NC:> after you execute the CADDs command. At this point, continue your CVNC session.

Moving to CADDs with IC

To execute a single Explicit CADDs command within CVNC, enter IC, followed by the selected CADDs command and modifiers.

Please note: You cannot use the mouse to provide input within the IC command line. Use the ~ for this purpose.

For example,

```
NC:> IC LIST LAYER
$85$LIST LAYER
ACTIVE CONSTRUCTION LAYER: 0
VISIBLE LAYERS:
DRAWING: 1 0-254
VIEW: TOP 0-254
VIEW: FRONT 0-254
VIEW: ISO 0-254
$85$
NC:>
```

In this example,

1. P10 displays the JCF contents.
2. IC is executed, specifying CADDs LIST LAYER. The prompt changes, and the LIST LAYER information is displayed.
3. The return of the NC : > prompt indicates CVNC is ready for more input.

Please note: To execute a series of CADDs commands, use RESU.

Entering Multiple CADDs Commands — RESU

Moving to CADDs with RESU

To move to CADDs from CVNC, execute RESU. The prompt changes to \$nn\$. (This environment uses the VNTAB.CAMNCPROC verb/noun table.)

At this prompt, you can execute CADDs commands. (CVNC, however, prevents you from executing any CADDs commands that might create inconsistencies between the CADDs and JCF databases.)

Please note: RESU suspends (not ends) your currently active CVNC session. You cannot begin a CVNC process of another type in parallel to the one suspended. Return to CVNC before exiting or quitting your CVNC session or CADDs part.

Returning to CVNC with RESUME NCPROC

To return to CVNC at the point you left it with RESU, execute RESUME NCPROC. The prompt changes to NC: >. For example,

```
NC:> P
.Top.
1 > [JCF CVNC MILL HEADER LINE] <
1 DATUM 'TOP'
1 HOMEPT FROM X -2.1 Y -24 Z 0 ;
1 DEFTOOL '1/4-MILL' MILL7 DIA .25
CORNER 0 HEIGHT 5 GAGE 5
1 DEFTOOL 'DRILL' DRILL DIA 1 GAGE 5
NC:> RESU
NC:> $nn$ECHO LAY 1
NC:> $nn$RESUME NCProcess
CPLANE ACTIVATED
NC:>
```

In this example,

1. The RESU command is executed. The prompt changes to \$nn\$.
2. The CADDs ECHO LAYER command is executed.
3. The CADDs RESUME NCPROC command is executed. The prompt changes back to NC: >.

Please note: To execute a single CADDs command, use ~ or IC.

Automating JCF Input

This chapter shows you how to efficiently perform repetitive command sequences using the command and execute file facilities CVNC provides.

- CVNC Command and Execute Files
- Using Command Files
- Using Execute Files

CVNC Command and Execute Files

Performing Repetitive Functions

Many part programming command sequences are used repetitively throughout a typical machining job, such as:

- Moving to the home point.
- Turning the coolant or spindle speed on and off.
- Configuring the machine.
- Changing tools.

CVNC provides three facilities to efficiently set up command sequences to perform these tasks:

- Command files
- Execute files
- CVMAC macros

Please note: This chapter includes directions for setting up and using command and execute files. For CVMAC macro explanations, see *Customizing CVNC*.

To help you determine which facility is best suited for your needs, the table below compares command and execute file features.

Command Files	Execute Files
Commands are processed, but <i>not</i> displayed on the screen.	Each command is displayed on the screen as it is processed.
Commands do not become a permanent part of the JCF.	Commands become a permanent part of the JCF.
Command files may contain only CVNC commands.	Execute files may contain CVNC and CADDs commands.
Commands can be displayed as level 2 commands.	Commands are displayed as level 1 commands.
Create command files while you are in a JCF or with a UNIX editor.	Create execute files with a UNIX editor.
Store command files in a directory of your choice or in a default directory specified by CVNC.	Store execute files in a default directory, specified by CVNC.

Choosing Which Facility to Use

You can use both command files and execute files to perform repetitive functions. For example, you can use either a command file or an execute file to set up the parameters to configure a machine tool. Here are sample reasons why you may choose one method over the other.

Command Files — Pros and Cons: For an experienced user, setting up machine tool configuration within a command file creates a faster execution method (because the commands within the file are not displayed as they are processed). The commands within the file can be accessed, however, using the IN or INX command (see Chapter 7).

Using command files, as opposed to entering the commands directly into the JCF, can decrease the JCF length.

Execute Files — Pros and Cons: For a less experienced user, setting up the machine tool configuration within an execute file provides more online communication as the process takes place (because the commands within the file are displayed as they are processed).

Also, because the commands become a permanent part of the JCF, the logic may be more readily visible to another person accessing the JCF.

Using execute files, as opposed to command file, can increase the JCF length.

Command and Execute File Commands

The following commands are described in this chapter:

- / (Locate string) — Locates a specified string in the JCF at any nested level.
- [(Block) — Selects the commands to be placed within a command file.
- CALL — Accesses a command file from the JCF.
- CMFIL — Specifies the command file name.
- COMLIB — Accesses CVNC command files stored in libraries.
- EXEC — Accesses an execute file from your JCF.
- READ — For interactive prompting.
- W (Write) — Stores the command file in a library.

Please note: Because CALL, CMFIL, COMLIB, and READ are used in conjunction with command file creation, they are described here. These commands are not CVNC editor commands, however.

Using Command Files

Storing Command Files

Stores command files in a library you select for them with the COMLIB command or a default library. Command files can contain only CVNC commands.

Use the COMLIB command to specify the name(s) of a library of command files. If you are creating the command file while you are in the JCF, specify the library name where the command file is to be stored. When you write the command file to the library, the library name is created. Command files must be stored under an `_bcd` directory.

If you are creating the command file offline,

1. Create the command file library in UNIX.
2. Specify this library within the JCF with the COMLIB command before you call the command file.

By default, command files created online (from within a JCF) are stored in your CVPATH `cam/application/com/lib/_bcd` directory, where `application` is replaced by the CVNC application abbreviation.

For	Specify
CVNC-M2	ncmill
CVNC-M3	nc3ax
CVNC-M5	nc5ax
CVNC-P2	ncfab
CVNC-T2	ncturn

You can also specify these directories to store offline created command files.

Command File Contents

Command files can contain any acceptable CVNC command sequence. They cannot include CVNC editor commands (the commands described in this book, with a few exceptions, as noted).

Command File Creation Methods

Create a command file

- Online from your JCF.
- Offline at UNIX level.

Procedures for each method are described next.

Creating Online Command Files

Here are the steps to create a sample online command file. This file is stored in the default library, prompts for the tool name, and changes the tool.

1. Select a unique name for the command file, eight or fewer characters, and enter the name with the CMFIL command. For example, to create a command file named GETTOOL,

```
CVNC MILLING
Entering NEW JCF 'SAMPLE'
NC:> DATUM 'TOP'
CPLANE ACTIVATED
  Setting tool location system variables:
#CURLOC is now X 0.0 Y 0.0 Z 0.0 in active cplane space.
NC:> HOMEPT FROM X -2 Y -3 Z 0;
NC:> CMFIL GETTOOL
```

2. Enter each CVNC command to appear in the command file.

```
NC:> READ "TOOL NAME?" TOOLNAM
NC:> READ TOOLNUM
NC:> CHGTOOL 1 TOOLNAM
```

3. Signal the command file end with the [(Block) command.

```
NC:> [
```

4. Enter P12 to display the JCF.

Please note: Note the following: [BLOCK] and [EOF] lines now precede and follow the command file lines, the CMFIL command line is within square brackets ([CMFIL GETTOOL]), and the lines within the command file are at the second nested level (shown by a 2 in the left margin).

```
NC:> P12
1          HOMEPT FROM X -2 Y -3 Z 0;
1          [BLOCK]
2          [CMFIL GETTOOL ]
2          READ "TOOL NAME?" TOOLNAM
2          READ TOOLNUM
2          CHGTOOL 1 TOOLNAM
2          [EOF]
1          > [DONE] <
```

5. Move back to the BLOCK line.

```
NC:> B/BLOCK
2<   ^   [EOF]
2     ^   CHGTOOL 1 TOOLNAM
2     ^   READ "TOOL NAME?" TOOLNAM
2     ^   READ TOOLNUM
1     ^>  [BLOCK] <
        ^---^
```

6. Change BLOCK to CALL name, where name is replaced with the command file name.

```
NC:> C/BLOCK/CALL GETTOOL
1     >  [CALL GETTOOL] <
```

7. Enter P10 to show the changes to the JCF.

```
NC:> P10
1     [JCF CVNC MILL HEADER LINE]
1     DATUM 'TOP'
1     HOMEPT FROM X -2 Y -3 Z 0;
1     >  [CALL GETTOOL] <
2     [CMFIL GETTOOL ]
2     READ "TOOL NAME?" TOOLNAM
2     READ TOOLNUM
2     CHGTOOL 1 TOOLNAM
2<    [EOF]
1     [DONE]
```

8. Execute the W (Write) command to write the command file out to the disk in the default library name.

```
NC:> W
2     CMFIL GETTOOL
2     READ "TOOL NAME?" TOOLNAM
2     READ TOOLNUM
2     CHGTOOL 1 TOOLNAM
1     >  CALL GETTOOL <
```

9. Enter P99 to show the changes to the JCF. (Note that the CALL command line is the only one remaining in the display.)

```
NC:> P99
.Top.
1     [JCF CVNC MILL HEADER LINE]
1     DATUM 'TOP'
1     HOMEPT FROM X -2 Y -3 Z 0;
1     >  CALL GETTOOL <
1     [DONE]
```

Please note: The CMFIL line is inactive within the JCF. CVNC does not allow you to delete this line. Therefore, you can always identify the command file. (You can delete the CMFIL line, however, using a UNIX editor).

Creating Offline Command Files

Here are the steps to create an offline command file stored in a specified library.

1. Create the library name to store the command file. For example,

```
%mkdir mycommandfiles
%cd mycommandfiles
%mkdir _bcd
%cd _bcd
```

2. Specify the command file name (eight or fewer characters) in the library to a UNIX editor.

```
%vi gettool
```

3. Enter each command to be in the command file on separate lines and write the file to the disk. The CMFIL line must be the first file line.

```
CMFIL GETTOOL
READ "TOOL NAME?" TOOLNAM
READ TOOLNUM
CHGTOOL 1 TOOLNAM
```

Please note: CALL does not recognize a text file as a command file unless the command file first line is CMFIL name. Always insert CMFIL name as the first line in your command file.

Accessing Command Files

To access a previously defined command file,

1. Enter the JCF.

```
CVNC MILLING
Entering OLD JCF 'SAMPLE'
```

Identify the command file library where the command file is stored with the COMLIB command, (unless the command file is stored in the CVPATH default directory (cam/application/com/lib/_bcd) and you have not changed COMLIB to any other name).

COMLIB specifies the command file library name(s). If more than one library is specified, CVNC searches each library sequentially to find the file.

Specify up to four command file libraries to be searched.

```
NC:> COMLIB MYCOMMANDFILES
```

2. Enter the command file name with the CALL command.

```
NC:> CALL GETTOOL
```

You can call as many command files within your JCF as desired. For example,

```
NC:> COMLIB STARTUPCOMMANDFILES
NC:> CALL STARTUP
NC:> CALL TOOLCONF
NC:> COMLIB TOOLCOMMANDFILES
NC:> CALL TOOLCHG
```

Modifying Command Files

To modify a previously defined command file from within your JCF, use the IN or INX commands and perform the desired edits within the file. (For more information on IN and INX, see Chapter 7.)

To modify a previously defined command file offline, use any UNIX editor.

Locating Lines within a Command File

To locate a specified text string in the JCF at any nested level, use the / (Locate Text String) command.

The / command does not alter the current line position in the JCF.

Using Execute Files

Execute File Storage

You must store execute files in the CVPATH directory `execute/_bcd`. Create this directory before you begin the execute file.

Execute File Contents

An execute file can contain

- CVNC commands.
- A combination of CVNC and CADDs commands.

To include CADDs commands in the execute file,

1. Execute the RESU and RESUME NCPROC sequence.
2. Use the ~ or IC.
3. Exit from CVNC and execute the CADDs commands.

If the file contains both CVNC and CADDs commands, the commands must be in a usable order that follows the proper conventions. For example, CVNC comment lines must be preceded by `#>`; CADDs comment lines must be preceded by `*`.

The execute file commands operate in the same manner as if they were entered manually.

Creating an Execute File

You must create execute files offline in UNIX. You cannot create an execute file while you are in your JCF.

Here are the steps to create an execute file that prompts for the tool name.

1. Change to your CVPATH directory and create the `execute/_bcd` directory to store your initial execute file. For example,

```
%mkdir execute
%cd execute
%mkdir _bcd
%cd _bcd
```

2. Specify a file name in this directory to a UNIX editor and create the execute file contents.

```
%vi gettool
#> Execute file GETTOOL
  READ "TOOL NAME?" TOOLNAM
```

Accessing Execute Files

To access a previously defined execute file, enter the EXEC (Execute File) command, followed by a comma (,) and the execute file name. (Do not include a blank space on either side of the comma.)

For example,

```
NC:> P9
1 [JCF CVNC MILL HEADER LINE]
1 DATUM 'TOP'
1 HOMEPT FROM X -2 Y -3 Z 0;
1 > [DONE] <
NC:> EXEC,GETTOOL
```

CVNC searches for the `execute/_bcd/gettool` file and executes and enters the file contents into the JCF.

```
1 #> Execute file GETTOOL
1 READ "TOOL NAME?" TOOLNAM
1 > [DONE] <
```

The system reads execute files as if they are being entered manually. Except for CVNC editor commands and CADDs commands, the execute file contents become a permanent part of the JCF.

Please note: EXEC performs in the same manner as the CADDs EXECUTE FILE command. (For more information on EXECUTE FILE, see CADDs online documentation.)

Modifying an Execute File

To modify an execute file, use any UNIX editor.

Decomposing JCF Lines

This chapter describes how to break down high-level motion commands, command files, or macros to their component parts.

- Using Command Decomposition - IN, INX,], and OUT
- Decomposing Motion Commands
- Decomposing Command Files
- Decomposing CVMAC Macros

Using Command Decomposition - IN, INX,], and OUT

You can break down a high-level motion command, command file, or a macro to its component parts.

Decomposition — IN

Use the IN or INX command to examine individual commands within

- High-level motion commands, such as PROFILE, POCKET, or CUT ENTITY.
- Command files.
- CVMAC macros.

For example, you can

1. Go into a profiling operation.
2. Analyze the individual cuts.
3. Make changes to the JCF.

IN and INX enter the next nested level of command files. For example, these commands decompose composite commands into feed rate changes and individual moves, such as linear and arc motion.

IN and INX perform the same function, except

- IN requires that you execute the command first.
- INX executes commands that have not been executed before going into them.

Please note: Command decomposition is not available with CVNC-P2.

Recomposition — OUT

Once you have examined and changed (if necessary) any information within the individual commands, return to the next upward level by executing the OUT command.

To discard changes made to decomposed command lines, execute the] (Mend) command.

Decomposing Motion Commands

Using IN

Decompose high-level motion commands such as PROFILE by executing the IN command. For example,

```
NC:> PROFILE TO $L1 $L2 $L3 $L4;
NC:> IN
1    >[PROFILE TO $L1 $L2 $L3 $L4]<
2    [#>]
2    CUT XABS 1.5 YABS 1.5 Z 0.4
2    CUT XABS 3.0 YABS 1.5 Z 0.0
2    .
2    .
2    .
2    [EOF]
1    [DONE]
```

When you decompose a motion command, the decomposition appears in one or both of the following formats:

- FEED/CUT
- Motion

FEED/CUT format lists a FEED statement, followed by one or more CUT statements. The following commands appear in this format:

- CUT
- CONNECT
- POSITION
- STEPIN
- STEPOUT
- PROFILE

Motion format lists motion statements that automatically adjust feed rates during output generation. The following commands appear in this format:

- MOVE
- APPROACH
- PLUNGE
- RETRACT
- CLEAR

FEED/CUT Format Example

In the following example, CUT ENTITY decomposes into FEED/CUT statements:

```
1 FEED APPROACH 10
1 FEED RETRACT 20
1 FEED CLEAR 30
1 FEED CUT 40
1 FEED PLUNGE 50
1 FEED CONNECT 60
1 PLANE ZAPPR 2
1 PLANE ZWORK -2
1 PLUNGE XINC -4
1 CUT TO $L1;DIRLOC X 5.27586 Y 1.42887 ;
1 [CUT ENTITY CHECK TO $L2 ;CCLW ON $L3 ;]
2 > [#>] <
2 FEED IPM 40.0000 (CUT feed rate)
2 CUT XABS 6.4526 YABS 2.2845 ZABS -2.0000
2 CUT XABS 6.4526 YABS 5.7845 ZABS -2.0000
2< [EOF]
```

In the following example, PROFILE decomposes into FEED/CUT statements:

```
1 FEED CUT 40
1 FEED PLUNGE 50
1 FEED CONNECT 60
1 HOMEPT FROM X 1 Y 1;
1 [PROFILE $L1 ...;]
2 > [#>] <
2 FEED IPM 40.0000 (CUT feed rate)
2 CUT XABS 2.4526 YABS 4.2845 ZABS 0.0000
2 CUT XABS 4.4526 YABS 4.2845 ZABS 0.0000 22 CUT
2 .
2 .
2< [EOF]
```

Motion Format Example

In the following example, APPROACH decomposes to a single motion statement:

```
1 [APPROACH TO $L1;]
2 > [#>] <
2 APPROACH XABS 9.0000 YABS 7.000 ZABS 0.000
  (APPROACH feed rate)
2< [EOF]
```

Combined FEED/CUT and Motion Format Example

The following TURN command decomposes into FEED/CUT and motion statements. Note that the STEPIN feed rate is set at RAPID; this generates a MOVE statement in the decomposition.

```
1 FEED STEPOUT 20
1 FEED RETRACT 30
1 FEED PLUNGE 50
1 FEED POSITION 60
1 FEED CLEAR 70
1 FEED CUT 80
1 FEED APPROACH RAPID
1 FEED STEPIN RAPID
1 [TURN $L3 $L4 $L5 $L6 $L7 $L8 $L9 $L10; MATL $L2 $L1;MAXDEPTH
.8]
continue [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 (CUT feed rate)
2 CUT XABS 0.5000 YABS 5.7000 ZABS 0.0000
2 CUT XABS 0.5000 YABS 6.0000 ZABS 0.0000
2 .
2 FEED IPM 60.0000 (POSITION feed rate)
2 CUT XABS 9.1000 YABS 2.2000 ZABS 0.0000
2 MOVE XABS 9.1000 YABS 0.5000 ZABS 0.00002
2 FEED IPM 80.0000 (CUT feed rate)
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 (STEPOUT feed rate)
2 CUT XABS 6.5000 YABS 1.4000 ZABS 0.0000
2 < [EOF]
```

The following POCKET decomposes into FEED/CUT and motion statements:

```
1 FEED CUT 40
1 FEED PLUNGE 50
1 FEED CONNECT 60
1 PLANE ZAPPR 5
1 PLANE ZWORK -2
1 [POCKET $L1 $L4 $L3 $L2 ;]
2 > [#>] <
2 APPROACH XABS 4.452 YABS 4.284 ZABS 5.0000
  (APPROACH feed rate)
2 PLUNGE XABS 4.4526 YABS 4.2845 ZABS -2.0000
  (PLUNGE feed rate)
2 FEED IPM 40.0000
2 CUT XABS 2.4526 YABS 4.2845 ZABS -2.0000
2 CUT XABS 4.4526 YABS 4.2845 ZABS -2.0000
2 FEED IPM 60.0000 (CUT feed rate)
```

```
2 CUT XABS 4.9526 YABS 4.2845 ZABS -2.0000
2 FEED CUT IPM 40.0000 (CUT feed rate)
2 .
2< [EOF]
```

Special Considerations with TRANSLATE, ROTATE, and MIRROR

When you are performing command decomposition in conjunction with the TRANSLATE, ROTATE, or MIRROR commands, additional considerations apply. For more information, see *CVNC System User Guide and Menu Reference*.

Decomposing Command Files

Decomposing Command Files with IN (or INX)

To decompose a command file, position to the CALL statement within the JCF and execute the IN command. For example,

```
NC:> IN
NC:> T99
1 >[CALL PLATE] <
2 [CMFIL PLATE]
2 SPEED OFF
2 COOLANT OFF
2 MOVE HOME
2 READ 'TOOL NAME?' TOOLNAM
2 READ 'TOOL NUMBER?' TOOLNUM
2 CHGTOOL TOOLNUM TOOLNAM
2 [EOF]
1 [DONE]
```

You can now edit the command file. For example,

```
NC:> SYN
NC:> F/EOF
NC:> SPEED ON
NC:> COOLANT ON
NC:> B/CALL
NC:> ON
NC:> T99
```

In this example,

1. SYN turns off command execution to perform edits quickly. (This is an optional step.)
2. F/EOF moves forward to the EOF marker.
3. SPEED ON and COOLANT ON turn on the spindle and coolant.
4. B/CALL moves back to the CALL command.
5. ON turns on command execution.
6. T99 displays the JCF.

The modified command file appears as:

```
1 >[CALL PLATE] <
2 [CMFIL PLATE]
2 SPEED OFF
2 COOLANT OFF
2 MOVE HOME
```

```
2 READ 'TOOL NAME?' TOOLNAM
2 READ 'TOOL NUMBER?' TOOLNUM
2 CHGTOOL TOOLNUM TOOLNAM
2 SPEED ON
2 COOLANT ON
2 [EOF]
1 [DONE]
```

Next, use the W (Write) command to overwrite the previous command file contents.

You must be positioned at a disabled CALL command (in square brackets) to use W (Write). CVNC reinstates the CALL command and writes the second-level block to the directory specified by the last COMLIB command. For example,

```
NC:> W
      =usr/command/files/_bcd/plate already exists
      TYPE "OK" TO overwrite
      OK
```

The edited command file now contains SPEED ON and COOLANT ON commands (as shown above), and the JCF looks like this:

```
1      >CALL PLATE<
1      [DONE]
```

Please note: When you use the IN (INX) command to expand an online command file and make changes to the JCF, the] (Mend) command restores the original command file. Thus, you can discard your changes. To save the edits made to a command file online, use the W (Write) command.

Decomposing CVMAC Macros

Decomposing Macros with INX

Use the INX command to decompose a macro. For example,

```
NC:> P99
.Top.
1 [JCF CVNC MILL HEADER LINE]
1 PLANE ZWORK 4
1 NCGROUP CUP1 ENT $C2 ;
1 HOMEPT FROM X 0 Y 0 Z 4 ;
1> DPOCK 'CUP1' ZSTEP 2 ZEND 0 <
1 [DONE]
.Bottom.
NC:> INX
Making 2 passes with Z increment = 2
Incremental stock = 0

POCKET START POINT IS:
X:-0.0 Y:0.0
PASS # 1 AT Z = 2
PASS # 2 AT Z = 0
NC:> P99

1 [JCF CVNC MILL HEADER LINE]
1 PLANE ZWORK 4
1 NCGROUP CUP1 ENT $C2 ;
1 HOMEPT FROM X 0 Y 0 Z 4 ;
1 [DPOCK 'CUP1' ZSTEP 2 ZEND 0 ]
2> [CMFIL] <
2 POCKET CUP1; QUERY
2 PLANE ZWORK 2
2 MOVE XABS -5.52943107967E-18 YABS 0.011102230246252
2 POCKET CUP1;
2 MOVE XABS -5.52943107967E-18 YABS 0.011102230246252
2 PLANE ZWORK 0
2 POCKET CUP1;
2 PLANE ZWORK 4
2< [EOF]
1 [DONE]
.Bottom.
NC:>
```

In this example, the DPOCK macro is decomposed.

Decomposing Macros with IN

You can go into a macro (including a macro in a JCF being reentered) using the IN command only if it is the last macro executed in the JCF.

Please note: For more information on CVMAC macros, see *Customizing CVNC*.

Using CVNC Search and Editing Facilities

This chapter tells you how to perform repetitive or complex editing tasks with CVNC star buffers and metacharacters.

- Search and Editing Facilities
- Using Metacharacters
- Using Star Buffers
- Combining Metacharacters and Star Buffers

Search and Editing Facilities

You can perform repetitive or complex editing tasks with CVNC star buffers and metacharacters.

Star Buffers

Star buffers allow you to store groups of CVNC editor or application commands in sequence and then call and execute them in your JCF.

You can:

- Define up to 10 separate star buffers for each JCF.
- List buffer information and display the contents of a specified buffer.
- Redefine or modify a star buffer at any time.

Metacharacters

Use metacharacters within text strings to search for and replace other text strings in the JCF. The following CVNC editor commands use text strings:

- + (Append)
- B (Backward)
- C (Change)
- F (Forward)

Text strings can be divided into search strings and input strings.

- Search strings are strings you direct the CVNC editor to locate in the JCF.
- Input strings are strings with which to replace a current string or to add to a current string.

Within these search or input strings, certain characters or character combinations have a special function, and are known as metacharacters.

Star Buffers and Metacharacters

Star buffers allow you to edit the JCF automatically without having to manually move backward and forward (using the B and F commands) to the command line to be edited.

With metacharacters, you can use star buffers to perform such tasks as globally changing feed rates.

Using Metacharacters

Metacharacter Symbols

Use metacharacters to search for and replace text strings within the JCF. Metacharacter symbols are described in the following table.

Symbol	Function
!	Matches any character.
”	Matches any uppercase alphabetic character A to Z.
%	Causes a character (except ^) to match one or more occurrences of itself until the match fails.
==	Matches the last string located by either F (Forward) or B (Backward).
?	Matches any numeric character 0 to 9.
[ABC]	Indicates a choice of one character among those within square brackets ([]) for a match. Characters can be numeric, text, metacharacters, or symbols.
	Causes a metacharacter or nonmetacharacter to match 0 or more (up to 80) of that character or metacharacter.
^	Causes the next character to be treated as itself, thus disabling its metacharacter properties.
~	In search strings, represents a line break character. In replacement strings, copies characters from the working string into the replacement string. (Use the TW (Type working string) command to display the working string.)

Metacharacter Examples

Here are examples illustrating each metacharacter.

Changing a Text String : To change
CAM.NCMILL.GRAMMAR.LOCAL.MILLING to TEST, enter

```
C/'!\\'/'TEST'
```

Interpretation: C/'(zero or more of any character)'/TEST

Searching for Consecutive Strings: To find the first two consecutive RETRACT commands, enter

```
F/RETRACT~== Interpretation: F/RETRACT (line break) (what F/RETRACT found)
```

Searching and Changing an Ending String : To locate a line ending XYLOC
X7.1 Y14.6 Z2.0
X12.6 Y11.31 Z2.0; and change it to XABS 10, enter

```
F/XYLOC!%
```

Interpretation: F/XYLOC (as many and any character as available)

```
C /==/XABS 10
```

Interpretation: C/(what F/XYLOC!% found)/XABS 10

Searching and Moving a String : To find the next XABS or YABS and append to the next line, enter

```
F/[XY]ABS F  
+/==
```

Interpretation: F/(X or Y)ABS append/(what F/[XY]ABS found)

Searching for Consecutive Identical Strings: To find two consecutive lines where the second line contains all the characters of the first, enter

```
F/!%~==
```

Interpretation: F/(as many of any character as can be found [whole line]) (line break) (what F/!% found)

Searching for a Discrete String: To find the first preceding FEED command that does not specify a motion class (CUT, APPROACH), enter

```
B/FEED \?
```

Interpretation: B/FEED(zero or more “spaces” followed by a numeric)

Searching for an Exclamation Point (!): To find an exclamation point, enter

```
F/^!
```

Interpretation: F/(exactly!)

Searching for Two Equal Signs (==): To find two equal signs, enter

F/^=^=

Interpretation: F/(exactly=)(exactly=)

Locating a String : To stop three lines from the file end, enter

F/~~~DONE

Interpretation: F/(nothing) (line break)(nothing) (line break)(nothing) (line break) DONE (Note that “nothing” in a search string matches “before the first character in the line,” which can always be found. Therefore, it matches any line.

Using Star Buffers

Star Buffer Procedure

To create and use a star buffer within your JCF,

1. Determine what commands the star buffer should contain.
2. Create the star buffer.
3. Within the JCF, execute the star buffer.

Please note: Star buffer contents are removed if the JCF binary versions are deleted. Create frequently used star buffers using an execute file, to ensure that they can be recreated.

Star Buffer Commands

The following table lists and describes star buffer commands.

Command	Function
*n	Initiates a new star buffer
E*n	Initiates editing of a previously defined star buffer.
FILE*	Saves changes made to a star buffer during an editing session.
I*	Inserts CVNC editor commands into the star buffer.
IL	Inserts application commands into the star buffer.
L*	Displays a list of star buffers associated with the JCF.
QUIT*	Discards changes made to a star buffer during an editing session of a previously defined star buffer.
T*n	Displays the specified star buffer contents.
X*n	Executes a specified star buffer within the JCF.

Creating a Star Buffer — *n

To create a new star buffer:

1. Execute *n, replacing n with an integer (0 to 9).
2. The prompt changes to I*n>, indicating you can now enter CVNC editor commands into the n star buffer.
3. Enter each selected CVNC editor command on a separate line.
4. Enter each selected application command on a separate line, prefixed with IL (for example, IL COOLANT OFF).
5. When all the CVNC editor and application commands are entered, press RETURN twice to return to the NC:> prompt.

For example

```
NC:> *1
I*1>F/IPM
I*1>C/IPM/MMPM
I*1>
NC:>
```

This example creates a star buffer 1.

The CVNC editor commands within the star buffer instruct CVNC to search the JCF for IPM and change this string to MMPM.

An extra RETURN at the end of the star buffer returns to the NC:> prompt.

Please note: Reopening a previously defined star buffer with the *n command causes the original star buffer contents to be overwritten.

Listing Star Buffers — L*

To list star buffers associated with a JCF, use L*. For example,

```
NC:> L*
*buffer 0 is undefined
*buffer 1 contains 3 commands
*buffer 2 is undefined
*buffer 3 is undefined
*buffer 4 is undefined
*buffer 5 is undefined
*buffer 6 is undefined
*buffer 7 is undefined
*buffer 8 is undefined
*buffer 9 is undefined
```

NC:>

In this example, L* displays a list of previously defined and undefined star buffers.

Displaying Star Buffers — T*n

To display the contents of a specified star buffer, execute T*n, replacing n with the index number 0 to 9 of the star buffer to be displayed. For example,

```
NC:> T*3
      F/PROFILE
      C/ON/TO
NC:>
```

This example shows the contents of star buffer 3.

Executing a Star Buffer — X*n

To execute the contents of a specified star buffer

1. Enter X*n, replacing n with an integer from 0 to 9 representing the star buffer index number.
2. Optionally, enter one of the following:
 - m Replacing m with an integer from 0 to 99 representing the number of times the star buffer is to be executed.
 - \$ To specify that the star buffer is to be executed repeatedly until reaching a condition that does not allow further execution (for example, [DONE]).

Please note: Note that a comma (,) separates n from the next modifier. For example,

```
NC:> T*1
      F/OPERATOR 1
      C/OPERATOR 1/OPERATOR 2
NC:> P8
Top.
1  [JCF CVNC MILL HEADER LINE]
1  > #>THIS JCF WAS CREATED BY OPERATOR 1 <
1  HOMEPT FROM X -2.1 Y -24 Z 0;
1  #>HOME POINT WAS DEFINED BY OPERATOR 1
1  [DONE]
.Bottom.
NC:> X*1,2
1  > #>THIS JCF WAS CREATED BY OPERATOR 1 <
1  > #>THIS JCF WAS CREATED BY OPERATOR 2 <
1  HOMEPT FROM X -2.1 Y -24 Z 0;
```

```
1 > #>HOME POINT WAS DEFINED BY OPERATOR 1<
^_--^
1 > #>HOME POINT WAS DEFINED BY OPERATOR 2 <
NC:> P9
.Top.
1 [JCF CVNC MILL HEADER LINE ]
1 #>THIS JCF WAS CREATED BY OPERATOR 2
1 HOMEPT FROM X -2.1 Y -24 Z 0;
1 > #>HOME POINT WAS DEFINED BY OPERATOR 2 <
1 [DONE]
.Bottom.
```

In this example, star buffer 1, which instructs CVNC to move to OPERATOR 1 and change it to OPERATOR 2, is executed two times ($X*1, 2$). The P (Page) command displays the resulting changes to the JCF.

X*n Options

You can execute the star buffer

- A specified number of times.
- Repeatedly, until reaching a condition that does not allow further execution (for example, [DONE]) .

You can initiate star buffer execution from within another star buffer.

Please note: Use the \$ option only with star buffers that cause the current JCF line to move forward or backward. This is because a star buffer containing no forward or backward motion (for example, a buffer containing only the P command) would execute infinitely. In this case, the star buffer executes once and a message stating that the buffer is not suitable for indefinite execution is displayed.

Editing a Star Buffer — E*n

To edit a previously defined star buffer, execute E*n and use CVNC editing commands to make changes or add new commands. For example,

```
NC:> T*3
      F/PROFILE
      C/ON/TO
NC:> E*3
.Top.
*      [Start * buffer
      + F/PROFILE +
E*3>F
      + C/ON/TO +
```

```
E*3>F
      + [End * buffer] +
.Bottom.
E*3>I*
I*3>C/REV//
I*3>
E*3>FILE*
NC:> T*3
      F/PROFILE
      C/ON/TO
      C/REV//
NC:>
```

In this example,

1. T*3 displays the contents of star buffer 3.
2. E*3 initiates editing of star buffer 3.
3. F moves the pointer forward one line until the end.
4. I* opens the star buffer to add new commands.
5. C/REV// is entered into the star buffer.
6. An extra RETURN returns to star buffer edit mode.
7. FILE* saves the edited star buffer contents.
8. T*3 displays the revised contents of star buffer 3.

Please note: Edits do not take effect until the star buffer is executed.

Inserting Additional Commands — I*n and IL

To insert additional CVNC editor or application commands in a previously defined star buffer,

1. Enter edit mode for the star buffer using E*n.
2. Use standard CVNC editing commands to position the current line markers at the line after which you want to insert new commands.
3. Enter I*.
4. Enter the CVNC editor commands to be inserted.
5. Enter each application command in the star buffer on a separate line, prefixed with IL (for example, IL COOLANT OFF).
6. Enter a RETURN to return to star buffer edit mode.

For example,

```
NC:> T*3
      F/PROFILE
      C/ON/TO
NC:> E*3
.Top.
      *      [Start * buffer
              + F/PROFILE +
E*3>F
              + C/ON/TO +
E*3>F
              + [End * buffer] +
.Bottom.
E*3>I*
I*3>IL COOLANT OFF
I*3>
E*3>FILE*
NC:> T*3
      F/PROFILE
      C/ON/TO
      IL COOLANT OFF
```

Saving Star Buffer Edits — FILE*

To save changes made to an edited star buffer, use FILE*. For example,

```
NC:> T*3
      F/PROFILE
      C/ON/TO
NC:> E*3
.Top.
      *      [Start * buffer
              + F/PROFILE +
E*3>F
              + C/ON/TO +
E*3>F
              + [End * buffer] +
.Bottom.
E*3>I*
I*3>C/REV//
I*3>
E*3>FILE*
NC:> T*3
      F/PROFILE
      C/ON/TO
      C/REV//
NC:>
```


In this example,

1. T*3 displays the contents of star buffer 3.
2. E*3 initiates editing of star buffer 3.
3. F moves the pointer forward to the star buffer end.
4. I* opens the star buffer to allow new commands to be added.
5. C/REV// is entered into the star buffer.
6. An extra RETURN returns to star buffer edit mode.
7. FILE* saves the edited star buffer contents.
8. T*3 displays the revised star buffer 3 contents.

Discarding Star Buffer Edits — QUIT*

To discard changes, use QUIT*. For example,

```

NC:> T*3
      F/PROFILE
      C/ON/TO
NC:> E*3
.Top.
   *      [Start * buffer
   + F/PROFILE +
E*3>F
   + C/ON/TO +
E*3>F
   + [End * buffer] +
.Bottom.
E*3>I*
I*3>IL COOLANT OFF
I*3>
E*3>QUIT*
NC:> T*3
      F/PROFILE
      C/ON/TO
NC:>

```

In this example,

1. T*3 displays the contents of star buffer 3.
2. E*3 initiates editing of star buffer 3.
3. F moves the pointer forward until the end of the star buffer is reached.
4. I* opens the star buffer to allow new commands to be added.
5. IL COOLANT OFF enters the COOLANT OFF command.
6. An extra RETURN is pressed to return to star buffer edit mode.

7. QUIT* discards the edited star buffer contents.
8. T*3 displays the original star buffer 3 contents.

Combining Metacharacters and Star Buffers

Star Buffer and Metacharacter Example

This example combines star buffers and metacharacters to globally change feed rate units from IPMs to MMPMs in a JCF.

```

NC:> P15
1 > [JCF CVNC MILL HEADER LINE] <
1 DATUM 'TOP'
1 HOMEPT FROM X -878.13 Y 1759.95 Z 5;
1 FEED APPROACH IPM 500
1 FEED PLUNGE IPM 300
1 FEED CUT IPM 1000
1 FEED RETRACT IPM 2000
1 SPEED RPM 6000
1 RETRACT
1 MOVE XYLOC X -1140.16 Y 1623.27 ;
1 APPROACH
1 PLUNGE XYLOC X -1034.94 Y 1623.27 ;
1 [DONE]
.Bottom.
NC:> *1
I*1>F/FEED!%
I*1>C/IPM/MMPM
I*1>
NC:> X*1,$
1 DATUM 'TOP'
1 > FEED APPROACH IPM 500 <
  ^-----^
1 > FEED APPROACH MMPM 500 <
1 > FEED PLUNGE IPM 300 <
  ^-----^
1 > FEED PLUNGE MMPM 300 <

1 > FEED CUT IPM 1000 <
  ^-----^
1 > FEED CUT MMPM 1000 <
1 > FEED RETRACT IPM 2000 <
  ^-----^
1 > FEED RETRACT MMPM 2000 <
Error on line > F/FEED!% *Buffer *1-terminated with 2-lines
pending.Text string not found.
NC:> P15
.Top.
1 > [JCF CVNC MILL HEADER LINE] <
1 DATUM 'TOP'
1 HOMEPT FROM X -878.13 Y 1759.95 Z 5; 1
  FEED APPROACH MMPM 500
1 FEED PLUNGE MMPM 300
1 FEED CUT MMPM 1000
1 FEED RETRACT MMPM 2000

```

```
1    SPEED RPM 6000
1    RETRACT
1    MOVE XYLOC X -1140.16 Y 1623.27 ;
1    APPROACH
1    PLUNGE XYLOC X -1034.94 Y 1623.27 ;
1    [DONE]
.Bottom.
NC:>
```

In this example,

1. Star buffer 1 is created. CVNC editor commands within it instruct CVNC to move forward to any FEED command line and change IPM to MMPM.
2. The star buffer executes until an error condition occurs (it can find no other occurrences of FEED).

Displaying Search Strings (TW)

To display the last search string located (usually by the F or B command), execute TW. For example,

```
NC:> /POCKET
8 > POCKET $L1 $L2 $L3 $L4 ; <
Command terminated with 2-lines of JCF pending.
NC:> TW
POCKET
NC:>
```

In this example, TW is executed, displaying the last text string located by the CVNC editor. (TW also displays what the metacharacter == represents. This is helpful when you are creating star buffers as aids in checking for programming errors.)

Index

A

A (Again) command
 procedure 4-13
Append (+) command
 procedure 4-10

B

B (Backward) command
 procedure 3-4, 3-5
BR (Brief) command
 procedure 3-10
Buffers
 star 8-1

C

C (Change) command
 procedure 4-7
CADDs
 accessing from CVNC 5-4
 returning from 5-4
CADDs Access Command (~)
 procedure 5-2
CALL command
 procedure 6-6
CMFIL (Command File) command
 procedure 6-5
COMLIB (Command Library) command
 procedure 6-4, 6-7
Command files 6-2

 accessing 6-7
 commands 6-3
 contents 6-4
 creating 6-5
 decomposing 7-7
 editing 6-8
 storing 6-4
Command lines
 repeating 4-13
Commands
 quick reference 1-7
Comment (#>) command
 procedure 4-9
Copy (=) command
 procedure 4-16
Create Star Buffer (*n) command
 procedure 8-8
Current line 1-3
Cut (4-19
CVNC overview
 editor 1-2
 advanced facilities (star buffers) 1-2
 commands (quick reference) 1-7
 icons 1-4
 popup menu 1-6
 exiting 3-8

D

D (Delete) command
 procedure 4-11
Decomposing
 command files 7-7

- commands 7-3
- macros 7-9
- Delete Mark (-) command
 - procedure 4-15
- Documentation, printing from Portable Document Format (PDF) file xiv

E

- E (Edit) command
 - procedure 4-4
- E*n (Edit Star Buffer) command
 - procedure 8-10
- ED (Edit Digitize) command
 - procedure 4-6
- Editor 1-2
- EX (Execution) command
 - procedure 3-22
- EXEC (Execute File) command
 - procedure 6-10
- Execute files 6-2
 - accessing 6-10
 - contents 6-9
 - creating 6-9
 - editing 6-10
 - storing 6-9
- EXIT command
 - procedure 3-8

F

- F (Forward) command
 - procedure 3-4, 3-5
- FC (Forward Compare) command
 - procedure 3-14
- FILE* command
 - procedure 8-12
- FU (Full) command
 - procedure 3-11

G

- GR (Graphics Display)
 - procedure 3-11

H

- Help
 - ! command 2-3
 - assist line 2-2
 - online 2-3

I

- i (Information icon) 2-2
- I* (Insert into Star Buffer) command
 - procedure 8-11
- IC (Insert CADDs Command)
 - procedure 5-2, 5-3
- IL (Insert Command into Star Buffer) command
 - procedure 8-11

J

- JCF lines 4-2
 - copying and moving 4-2
 - cutting and pasting 4-19
 - deleting 4-3
 - canceling deletions 4-12
 - graphic entities 4-6
 - marking and unmarking 4-15
 - selecting and copying 4-16
 - text
 - appending 4-10
 - replacing 4-7
- JCFs
 - automating input 6-1
 - comment lines 4-9
 - comparing evaluations 3-14
 - controlling line execution 3-22
 - deleting lines 4-11
 - displaying
 - lines 3-2
 - tags and location data 3-9
 - editing 1-2
 - editing lines within 4-2
 - exiting 3-8
 - full and brief display options 3-10
 - graphic display options 3-11
 - locking 3-13
 - moving through 3-4

packing 3-24
 processing control 3-21
 quitting 3-8
 recovering old evaluations 3-16
 restoring 3-19
 saving 3-19
 securing 3-17
 syntax checking 3-21
 tags and location data 3-9
 unlocking 3-14

L

L* (List Star Buffers) command
 procedure 8-8
 Locate Text String (/) command
 procedure 6-8
 LOCK command
 procedure 3-13

M

Mark (.) command
 procedure 4-15
 Mend () command
 procedure 7-2
 Metacharacters
 examples 8-4
 symbols 8-4
 uses 8-2
 Motion commands
 decomposing 7-3

O

OFF command
 procedure 3-21
 ON command
 procedure 3-21
 OUT command
 procedure 7-2

P

P (Page) command
 procedure 3-2
 PACK command
 procedure 3-24
 PACK JCF command
 procedure 3-24
 Paste (>) command
 procedure 4-19
 Printing documentation from Portable
 Document Format (PDF) file *xiv*

Q

QUIT command
 procedure 3-8
 QUIT* command
 procedure 8-13

R

RECOVER command
 procedure 3-16
 Repeat Last Application Command (!!)
 procedure 4-13
 RESTART command
 procedure 3-4, 3-6
 RESTORE command
 procedure 3-19
 RESU command
 procedure 5-4
 RESUME NCPROC command
 procedure 5-4

S

SAVE command
 procedure 3-19
 Search strings
 displaying 8-16
 SECURE command
 procedure 3-17
 SEE command

- examples 3-9, 3-10
- procedure 3-9
- Select (#) command
 - procedure 4-16
- Star buffers
 - commands 1-10, 8-7
 - creating 8-8
 - discarding edits 8-13
 - editing 8-10
 - executing 8-9
 - inserting command into 8-11
 - inserting into 8-11
 - listing 8-9
 - saving edits 8-12
 - typing 8-9
 - uses 8-2
- SYN command
 - procedure 3-21
- Syntax checking 3-21
- System
 - markers 1-3

- procedure 6-6

X

- X*n (Execute Star Buffer) command
 - procedure 8-9

T

- T (Type) command
 - procedure 3-3
- T*n (Type Star Buffer) command
 - procedure 8-9
- Text delimiters 1-3
- TW (Type Working String) command
 - procedure 8-16
- Typing
 - procedure 3-3
 - Type Star Buffer 8-9
 - Type Working String command 8-16

U

- UNDEL command
 - procedure 4-12
- UNLOCK command
 - procedure 3-14

W

- W (Write) command